

# Principes Mathématiques et Utilisations des Algorithmes Génétiques

Christophe Bontemps\*

18 Novembre 1995

## Résumé

Un algorithme génétique est un “*algorithme stochastique itératif*” qui opère sur des ensembles de points *codés*, à partir d’une *population initiale*, et qui est bâti à l’aide de trois opérateurs : *croisement*, *mutation*, *sélection*. Les deux premiers sont des opérateurs d’exploration de l’espace, tandis que le dernier fait évoluer la population vers les optima d’un problème. Nous détaillons ici les principes de fonctionnement de ces opérateurs ainsi que les fondements mathématiques sur lesquels ils reposent.

Ces méthodes ne se réduisent cependant pas à la simple recherche d’optima d’une fonction, et s’avèrent être également de puissants outils pour l’analyse de situations dynamiques complexes comme l’on en rencontre sur les marchés financiers, ou en théorie des jeux (voir Axelrod [6]). On peut ainsi modéliser les comportements d’agents ou de stratégies et examiner la survie et l’évolution de ceux-ci, c’est à dire l’émergence de certaines stratégies dominantes.

---

\*Un grand merci à Jean-Marc Alliot, Christophe Bisière, Nicolas Durand, Nathalie Lenoir et Eric Malin.

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Principes</b>	<b>6</b>
2.1	Principes généraux des algorithmes génétiques . . . . .	7
2.2	Codage et Opérateurs . . . . .	10
2.2.1	Coder ou ne pas coder? . . . . .	10
2.2.2	Gestion des contraintes . . . . .	11
2.2.3	Génération aléatoire de la population initiale . . . . .	12
2.2.4	Sélection . . . . .	12
2.2.5	Croisement . . . . .	16
2.2.6	Mutation . . . . .	18
<b>3</b>	<b>Améliorations classiques</b>	<b>20</b>
3.1	Introduction . . . . .	20
3.2	Scaling . . . . .	20
3.2.1	Scaling linéaire . . . . .	21
3.2.2	Scaling exponentiel . . . . .	21
3.3	Partage (sharing) . . . . .	23
3.3.1	Principe . . . . .	23
3.3.2	Partage par bouquets (Clustered sharing) . . . . .	25
3.4	Algorithmes génétiques et recuit simulé . . . . .	27
3.4.1	Introduction . . . . .	27
3.5	Recuit simulé . . . . .	27
3.5.1	Principe . . . . .	28
3.6	Croisement avec recuit . . . . .	28
3.6.1	Principe du croisement avec recuit simulé . . . . .	28
<b>4</b>	<b>Utilisation et exemples</b>	<b>29</b>
<b>5</b>	<b>Résultats théoriques</b>	<b>30</b>
5.1	Modélisation de l’algorithme génétique . . . . .	31
5.2	Description rapide de l’algorithme . . . . .	31
5.2.1	Mutation $X_k \longrightarrow Y_k$ . . . . .	32
5.2.2	Croisement $Y_k \longrightarrow Z_k$ . . . . .	32
5.2.3	Sélection $Z_k \longrightarrow X_{k+1}$ . . . . .	33
5.3	Modélisation . . . . .	33
5.3.1	Processus de fond ( $X_k^\infty$ ) . . . . .	34

5.3.2	Processus perturbé ( $X_k^l$ ) . . . . .	35
5.4	La théorie de Freidlin et Wentzell . . . . .	39
5.4.1	Principe . . . . .	39
5.4.2	Mesure invariante $\nu^\varepsilon$ . . . . .	41
5.4.3	Dynamique du processus. . . . .	42
5.5	Résultats de convergence . . . . .	43
5.5.1	Taille critique . . . . .	44
5.5.2	Vitesse de convergence . . . . .	45
5.5.3	En guise de conclusion . . . . .	46
<b>6</b>	<b>Perspectives</b>	<b>47</b>
<b>7</b>	<b>Appendice : Un exemple élémentaire.</b>	<b>48</b>

# 1 Introduction

Par définition même l'étude de l'économie se base sur l'optimisation, et les deux mots sont souvent associés. La sophistication des modèles économiques conduit de plus en plus souvent à des problèmes d'optimisation toujours plus complexes. Le seul contexte des modèles dynamiques nous fournit bon nombre de problèmes de résolution ou d'estimation. Les études économiques s'en trouvent limités, et le modélisateur doit alors incorporer des contraintes techniques pour faire face à ces problèmes. Cette situation est quelque peu paradoxale, si l'on considère l'augmentation des moyens et des puissances de calcul disponibles.

Un grand nombre de méthodes économétriques, comme les moindres carrés généralisés et le maximum de vraisemblance, reposent sur la maximisation d'une fonction souvent complexe. Parmi toutes les méthodes d'optimisation, les méthodes de gradient sont les plus exigeantes en matière de conditions nécessaires de convergence. Ce sont des méthodes d'optimisation locales, qui connaissent une grande popularité. Pour le maximum de vraisemblance, Cramer [?], recense les désagréments possibles lors de l'application de cette méthode. L'algorithme de maximisation peut ainsi ne pas converger en un temps acceptable, il peut "s'éloigner" et donner des valeurs infinies pour certaines composantes des paramètres, il peut également boucler, et revenir sans cesse au même point, etc.... Ces algorithmes d'optimisation conventionnels<sup>1</sup> (de la famille du gradient ou de Newton-Raphson) sont des algorithmes "grimpeurs" qui se basent sur l'évaluation préalable du gradient, la pente, pour déterminer la direction de recherche de l'optimum. Goffe et al. [16] comparent cette situation à celle d'un aveugle cherchant le sommet d'une montagne. La connaissance du terrain passe alors uniquement par ses pieds. Pour peu que le terrain soit régulier et le point de départ bon, il atteindra le sommet. Toutefois ces deux conditions sont rarement simultanément réalisées. Avec beaucoup de chance, c'est à dire avec une très bonne sensibilité et beaucoup de points de départs, il pourra atteindre le sommet pour autant que ce dernier soit unique. De plus, les hypothèses sur les fonctions à optimiser sont souvent très fortes<sup>2</sup>, et ne sont pas vérifiées dans la pratique.

A l'inverse, les techniques de recherche aléatoires pures ne requièrent au-

---

<sup>1</sup>Les logiciels statistiques les plus utilisés comme GAUSS, SAS, RATS, utilisent des algorithmes plus ou moins évolués de ce type.

<sup>2</sup>fonction approximativement quadratique, unicité du maximum, de telle sorte qu'un maximum local est global, etc..

cune hypothèse particulière sur la fonction d'évaluation et explorent l'espace sans tirer partie des propriétés de la fonction objectif. Les algorithmes génétiques et le recuit simulé se situent entre ces deux extrêmes.

Une des particularités séduisantes des algorithmes génétiques et du recuit simulé, réside dans l'absence d'hypothèses particulière sur la régularité de la fonction objectif. Aucune hypothèse sur la continuité de cette fonction n'est requise, ses dérivées successives ne sont pas nécessaires, ce qui rend très vaste le domaine d'application de ces algorithmes. En outre ces algorithmes sont aisément parallélisables ce qui renforce leur efficacité dans le cadre de problèmes d'optimisation sur des espaces de dimension importante.

Les premiers travaux sur les algorithmes génétiques ont commencé dans les années cinquante lorsque plusieurs biologistes américains ont simulé des structures biologiques sur ordinateur. Puis entre 1960 et 1970, John Holland [18], sur la base des travaux précédents, développa les principes fondamentaux des algorithmes génétiques dans le cadre de l'optimisation mathématique. Malheureusement, les ordinateurs de l'époque n'étaient pas assez puissants pour envisager l'utilisation des algorithmes génétiques sur des problèmes réels de grande taille. L'ouvrage de Goldberg [17] qui décrit l'utilisation des algorithmes génétiques dans le cadre de résolution de problèmes concrets a permis de mieux faire connaître ces derniers et a marqué le début d'un nouvel intérêt pour ces techniques.

Le peu d'hypothèses requises sur la fonction objectif permet de traiter des problèmes très complexes. La fonction objectif peut ainsi être le résultat d'une simulation. On peut même imaginer, pour régler certains paramètres de l'algorithme génétique lui-même tels que la taille de la population, les différents pourcentages de croisement et de mutation, d'utiliser un algorithme génétique. La rapidité de convergence du premier devenant ainsi fonction d'évaluation du second.

Ces méthodes ne se réduisent cependant pas à la simple recherche d'optima d'une fonction, et s'avèrent être également de puissants outils pour l'analyse de situations dynamiques complexes comme l'on en rencontre sur les marchés financiers, ou en théorie des jeux (voir Axelrod [6]). On peut ainsi modéliser les comportements d'agents par des suites d'éléments binaires correspondant à des stratégies et examiner la survie et l'évolution de ces agents, c'est à dire l'émergence de certaines stratégies<sup>3</sup>.

Ces méthodes ne sont apparues que très récemment dans la littérature économique et économétrique. Pourtant les modèles économétriques actuels

---

<sup>3</sup>La stratégie "œil pour œil" survit ainsi très bien dans le cadre du dilemme du prisonnier répété (voir Axelrod [6], ou Petit-Singeot et Cazoulat [22]).

gènèrent bon nombre de problèmes de maximisation sur des espaces de dimensions grandissantes. Dorsey et Mayer [12] ont ainsi étudié récemment onze problèmes économétriques classiques en utilisant six procédures de maximisation différentes et leurs conclusions sont élogieuses envers les algorithmes génétiques. Andréoni et Miller [2] ont utilisé cet outil pour organiser des enchères, remplaçant les agents par des algorithmes adaptatifs basés sur des algorithmes génétiques. De leur étude, quoique fort restrictive, résulte une meilleure connaissance des procédures menant à l'équilibre de Nash dans différents types d'enchères.

Holland et Miller [19] proposent d'étendre encore le champ de ces techniques par l'utilisation d'Agents Artificiels Adaptatifs (AAA). Ils concluent leur article ainsi :

*“By executing these models (AAA) on a computer we gain (...) an opportunity to check the various unfolding behavior for plausibility, a kind of “reality check”. Whether or not agents behave in an optimal manner, the very act of contemplating such systems will lead to important questions and answers.”*

Nous proposons ici une présentation simple de ces nouveaux outils que sont les algorithmes génétiques et le recuit simulé. Notre but est d'expliquer les mécanismes généraux de ces algorithmes et d'en faire découvrir les usages et potentialités. La présente section est consacrée aux principes généraux qui animent ces algorithmes. Des illustrations mathématiques, économiques et économétriques sont présentées dans la section 4.. Nous proposons une synthèse des résultats théoriques les plus récents dans la section 5. Ces résultats sont apparus très tardivement, probablement à cause de la complexité induite par ces algorithmes et il faudra attendre 1993 pour qu'une démonstration complète et rigoureuse de convergence stochastique soit établie par R. Cerf [11]. La section 6 est consacrée aux perspectives qu'offrent ces outils dans les différents domaines de l'économie mathématique et de l'économétrie. En outre, nous donnons en appendice le traitement complet d'un exemple simple.

## 2 Principes

Les algorithmes génétiques sont des procédures qui s'inspirent des mécanismes de sélection naturelle et des phénomènes génétiques. Le principe de base consiste à simuler le processus d'évolution naturelle dans un environnement hostile. Ces algorithmes utilisent un vocabulaire similaire à celui de la génétique, cependant, les processus auxquels ils font référence sont beaucoup plus complexes.

On parlera ainsi d'individu dans une population. L'individu est composé d'un ou plusieurs chromosomes. Les chromosomes sont eux-mêmes consti-

tués de gènes qui contiennent les caractères héréditaires de l'individu. Les principes de *sélection*, de *croisement*, de *mutation* introduits dans ce cadre artificiel, s'appuient sur les processus naturels du même nom.

Pour un problème d'optimisation donné, un individu représente un point de l'espace d'état. On lui associe la valeur du critère à optimiser. L'algorithme génère ensuite de façon itérative des populations d'individus sur lesquelles on applique des processus de sélection, de croisement et de mutation. La sélection a pour but de favoriser les meilleurs éléments de la population, tandis que le croisement et la mutation assurent une exploration efficace de l'espace d'état.

## 2.1 Principes généraux des algorithmes génétiques

Le mécanisme consiste à faire évoluer, à partir d'un tirage initial, un ensemble de points de l'espace vers **le** ou **les** optima d'un problème d'optimisation. l'ensemble du processus s'effectue à taille de population constante, que nous notons  $N$ . Par analogie avec la génétique, on parle alors de générations successives. L'ensemble du processus s'effectue à taille de population constante, que nous notons  $N$ , de sorte que les générations successives comportent toutes  $N$  individus.

Afin de faire évoluer ces populations de la génération  $k$  à la génération  $k + 1$ , trois opérations (voir figure 1 ) sont effectuées pour **tous** les individus de la génération  $k$  :

- Une **sélection** d'individus de la génération  $k$  est effectuée en fonction du critère à optimiser ou plus généralement du critère d'adaptation au problème (fitness), on cherche ainsi à privilégier la reproduction des “bons” éléments au détriment des “mauvais”<sup>4</sup>.  
Des opérateurs d'exploration de l'espace sont ensuite utilisés pour “élargir” la population et introduire de la nouveauté d'une génération sur l'autre.
- L'opérateur de **croisement** est appliqué avec une probabilité  $P_c$  à deux éléments de la génération  $k$  (parents) qui sont alors transformés en deux nouveaux éléments (les enfants) destinés à les remplacer dans la génération  $k + 1$ .
- Certaines composantes (les gènes) de ces individus peuvent ensuite être modifiés avec une probabilité  $P_m$  par l'opérateur de **mutation**. Cette procédure vise à introduire de la nouveauté dans la population.

---

<sup>4</sup>Ces qualificatifs n'étant que relatifs à la population et à la fonction d'adaptation  $f$  considérés.

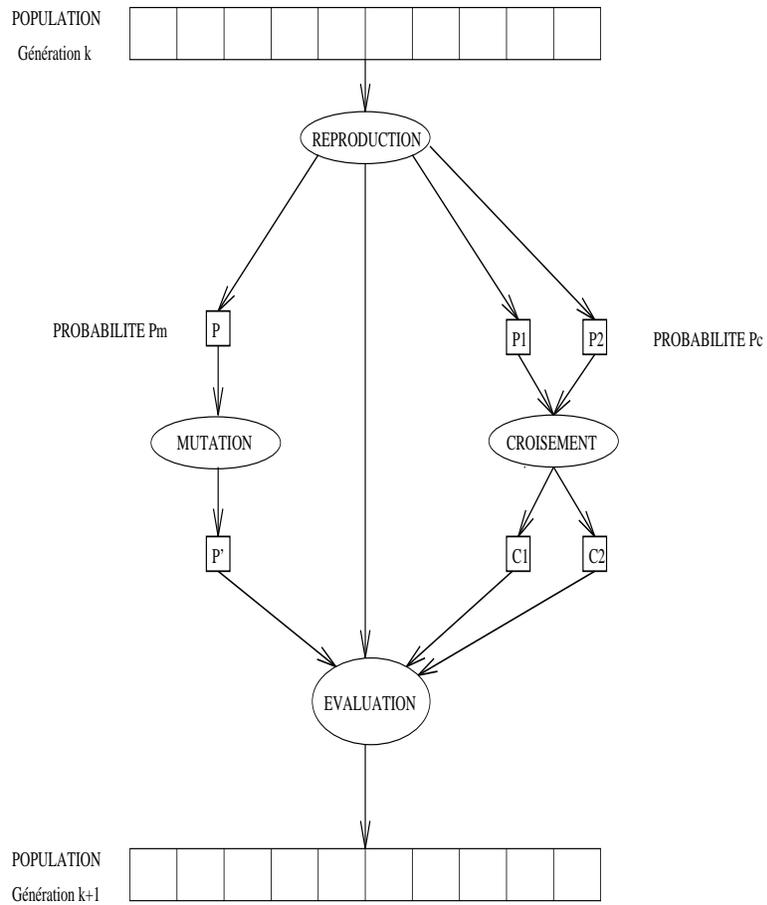


FIG. 1: Principe général des Algorithmes Génétiques

Cette procédure en trois points est ensuite renouvelée à taille de population constante. Les critères d'arrêts sont alors de deux natures :

- Arrêt après un nombre de générations fixé *a priori*. C'est la solution retenue lorsqu'un impératif de temps de calcul est imposé.
- Arrêt lorsque la population cesse d'évoluer ou n'évolue plus suffisamment rapidement, on est alors en présence d'une population homogène dont on peut penser qu'elle se situe à proximité du ou des optimums.

Il est à noter qu'à ce stade, aucune certitude concernant la bonne convergence de l'algorithme n'est assurée. Comme dans toute procédure d'optimisation l'arrêt est arbitraire, et la solution "en temps fini" ne constitue qu'une approximation de l'optimum.

Pour utiliser un algorithme génétique sur un problème particulier on doit disposer des cinq éléments suivants :

- Un principe de codage des éléments de l'espace admissible du problème, en éléments sur lesquels peuvent s'appliquer les trois opérateurs présentés ci-dessus. Ce codage intervient après une phase indispensable de modélisation mathématique du problème (voir figure ??).  
Le choix du codage des données dépend du problème traité et conditionne l'efficacité (vitesse de convergence, précision,...) de l'algorithme génétique.
- Un mécanisme de génération de la population initiale. Cette population initiale, qui sert de base aux générations futures, doit être *la plus hétérogène possible*.
- Une fonction d'utilité  $f$ , permettant de calculer l'adaptation de chaque élément au problème. Ce critère retourne une valeur de  $\mathbb{R}^+$  appelée *fitness*.
- Des opérateurs permettant de diversifier et d'améliorer la population d'une génération sur l'autre ainsi que d'explorer le plus largement possible l'espace admissible.
- Des paramètres dimensionnels : taille de la population, critère d'arrêt, probabilités de croisement ( $P_c$ ) et de mutation ( $P_m$ ).

Il s'agit donc d'un "algorithme stochastique itératif" qui opère sur des ensembles de points *codés*, à partir d'une *population initiale*, et qui est bâti à l'aide de trois opérateurs : *croisement*, *mutation*, *sélection*. Les deux premiers sont des opérateurs d'exploration de l'espace, tandis que le dernier fait évoluer la population vers les optima du problème. Nous détaillons dans la

section suivante les différentes phases de l’algorithme ainsi les principes de fonctionnement de ces opérateurs.

## 2.2 Codage et Opérateurs

### 2.2.1 Coder ou ne pas coder ?

Historiquement, les individus intervenant dans un algorithme génétique étaient codés sous forme de chaînes de bits<sup>5</sup>. Ce codage binaire contenant toute l’information nécessaire à la description d’un point dans l’espace d’état (voir également Alliot et Schiex [1]). Les opérateurs précités agissent alors sur les individus codés, c’est à dire sur de chaînes de bits.

A titre d’exemple, considérons le problème de maximisation d’une fonction  $f(x)$  définie sur le domaine  $[0, 1]$ , dont le maximum est atteint pour  $x^* = 0.5$ .

Pour traiter ce problème, on associe les points du domaine  $[0, 1]$  à une chaîne de bits  $V$  dont la longueur  $P$  déterminera la précision de résolution. La chaîne  $V$  sera donc composée de  $P$  éléments binaires  $V = (v_i)_{i=1,\dots,P}$  où  $v_i \in \{0, 1\}$ . Le point  $x_V$  correspondant sera défini par :

$$x_V = \sum_{i=1}^P v_i \cdot 2^{-(i-1)}$$

Ceci nous permet d’obtenir  $2^P$  éléments différents dans l’intervalle  $[0, 1]$ , ce qui nous donne une précision de  $\frac{1}{2^P-1}$ .

En examinant le codage au voisinage de 0.5, on constate que deux points très proches dans l’espace d’état peuvent être codés très différemment<sup>6</sup>.

En effet :

<i>Variable</i>	<i>Codage</i>
0.49999...	0111111111...
0.50000...	1000000000...

On évite cet inconvénient en utilisant un codage de Gray<sup>7</sup>.

Pour des problèmes d’optimisation dans des espaces de dimension supérieure, on concatène les chaînes de bits bout à bout. Par exemple, pour une

---

<sup>5</sup>Par analogie avec la biologie on parle également de “*chromosomes*”, les bits représentant les “*gènes*” composant ce chromosome.

<sup>6</sup>On utilise souvent la “*distance de Hamming*” comme mesure de la dissimilarité entre deux éléments de population pour le codage binaire. Cette mesure compte les différences de bits de même rang de ces deux sequences.

<sup>7</sup>Un code de gray possède la propriété de ne faire differer que d’un bit deux entier successifs.

fonction de deux variables  $z = f(x, y)$ , on code  $x$  et  $y$  sur leur domaine respectif puis on concatène  $x$  et  $y$  en une chaîne unique  $xy$ . Ce type de codage fonctionne bien mais présente l'inconvénient de perdre la structure du problème en fusionnant  $x$  et  $y$  dans une chaîne unique.

Il est également possible de ne pas coder les éléments de l'espace admissible du problème. Les opérateurs agissant directement sur les éléments de la population. Ainsi, les algorithmes génétiques utilisant des vecteurs réels étudiés par Golberg [?] et Wright [?] évitent ce problème en conservant les variables du problème dans le codage de l'élément de population sans passer par le codage binaire intermédiaire. L'exemple précédent serait codé à l'aide d'un vecteur à deux dimensions, on conserve ainsi la structure du problème dans le codage.

Nous verrons par la suite comment agissent les opérateurs sur des éléments codés sous forme de chaînes de bits, et sur des éléments réels.

### 2.2.2 Gestion des contraintes

La gestion des contraintes liées au problème est une tâche difficile et sensible pour laquelle l'utilisateur aura à arbitrer entre différentes techniques suivant son appréhension du problème. Ces contraintes sont de diverses natures et peuvent intervenir sur l'espace d'état (contraintes de signe, restrictions à un sous-espace, etc.), ou de manière plus complexe dans le problème lui même.

Dans le cas de contraintes sur l'espace dans lequel doit se faire la recherche, on pourra sélectionner les individus rapidement (sans avoir à les réévaluer) par différentes méthodes. Un individu "*hors champ*" pourra être :

- rejeté brutalement et remplacé par un autre individu tiré aléatoirement sur l'espace admissible ;
- ramené à la frontière la plus proche (principe du mur) ;
- reporté à la frontière diamétralement opposée à la frontière la plus proche (principe du tore).

Enfin il est bon de noter qu'il peut être préférable de garder des individus "*hors champ*" mais qui conservent une direction originale, plutôt que de confiner la population à un sous-espace.

Dans l'hypothèse où la gestion des contraintes ne peut se faire directement, les contraintes peuvent être incluses dans le critère à optimiser sous forme de pénalités. Ainsi, un individu qui viole une contrainte se verra attribuer une mauvaise fitness et sera donc éliminé, avec une forte probabilité, par le processus de sélection (voir section 2.2.4). Cette façon de gérer

les contraintes est difficile. En effet, inclure les contraintes dans la fonction d'évaluation peut se faire de diverses façons, et un "dosage" s'impose pour ne pas favoriser la recherche de solutions admissibles au détriment de la recherche de l'optimum ou inversement. On risque alors de fournir une solution admissible certes, mais éloignée de l'optimum.

### 2.2.3 Génération aléatoire de la population initiale

Comme dans tout problème d'optimisation, une connaissance de "bons" points de départ conditionne la rapidité de la convergence vers l'optimum.

Si la position de l'optimum dans l'espace d'état est totalement inconnue, il est naturel de générer aléatoirement des individus en faisant des tirages uniformes dans chacun des domaines associés aux composantes de l'espace d'état, en veillant à ce que les individus produits respectent les contraintes.

Si par contre, des informations *a priori* sur le problème sont disponibles, il paraît bien évidemment naturel de générer les individus dans un sous-domaine particulier afin d'accélérer la convergence.

Une nouvelle fois, les contraintes du problème pourront être incorporées (ou non) dans le tirage de la génération initiale.

Disposant d'une population d'individus non homogène, la diversité de la population doit être entretenue au cours des générations afin de parcourir le plus largement possible l'espace d'état, c'est le rôle des opérateurs de croisement et de mutation. Toutefois cette méthode diffère de la méthode de recherche aléatoire, puisque les générations successives doivent être évaluées et modifiées afin de converger, c'est ici qu'intervient l'opérateur de sélection.

### 2.2.4 Sélection

La sélection permet d'identifier statistiquement les meilleurs individus d'une population et d'éliminer les mauvais. On trouve dans la littérature un nombre important de principes de sélection plus ou moins adaptés aux problèmes qu'ils traitent. Les trois principes de sélection suivants ont retenu notre attention :

- *Ordonnancement*,
- *Roue de la fortune*,
- *Roue modifiée*, voir Golberg [?].

#### **Ordonnancement (Ranking)**

C'est le principe de sélection le plus simple, il consiste à attribuer à chaque individu son classement par ordre d'adaptation. Le meilleur (c'est à dire celui qui possède la meilleure fitness) sera numéro un, et ainsi de suite. On tire ensuite une nouvelle population dans cet ensemble d'individus ordonnés, en utilisant des probabilités indexées sur les rangs des individus. Cette procédure semble toutefois assez simpliste et exagère le rôle du meilleur élément au détriment d'autres éléments potentiellement exploitables. Le second, par exemple, aura une probabilité d'être sélectionné nettement plus faible que celle du premier, bien qu'il puisse se situer dans une région d'intérêt. Des procédures plus évoluées permettent de pondérer cette dominance des meilleurs éléments, c'est le cas des principes de roulette.

Le principe de la *Roue de la fortune* consiste à associer à chaque individu  $\theta_i$  une probabilité  $P_i$  proportionnelle à sa fitness  $f(\theta_i)$  dans la population. Cette probabilité pourra être calculée comme :

$$P_i = \frac{S(f(\theta_i))}{\sum_{j=1}^N S(f(\theta_j))}$$

où  $S$  est une fonction régulière et croissante, au sens large.

Chaque individu est alors reproduit avec la probabilité  $P_i$ , certains individus (les "bons") seront alors "plus" reproduits et d'autres (les "mauvais") éliminés.

**Remarque :**

Dans la pratique, il est aisé de tirer  $N$  individus, affectés de la probabilité  $P_i$ , parmi  $N$  avec remise. Pour cela, on associe à chaque individu un segment dont la longueur est proportionnelle à sa fitness (ou à  $S(f(\theta_i))$ , plus exactement). On reproduit ici le principe de tirage aléatoire utilisé dans les roulettes de foire<sup>8</sup> avec une structure linéaire. Ces segments sont ensuite concaténés sur un axe que l'on normalise entre 0 et 1 (voir figure ??). On tire alors un nombre aléatoire, de distribution uniforme entre 0 et 1, puis on "regarde" quel est le segment sélectionné. Avec ce système, les grands segments, c'est-à-dire les bons individus, seront plus souvent adressés que les petits, on privilégie ainsi les individus ayant une forte fitness au détriment des individus moins forts, tout en gardant une structure aléatoire, voir également l'exemple élémentaire détaillé en annexe..

---

<sup>8</sup>Les "roues de la fortune" que l'on rencontre dans les foires ont souvent des secteurs de tailles différentes suivant l'importance du lot. C'est ce principe qui est reproduit ici.

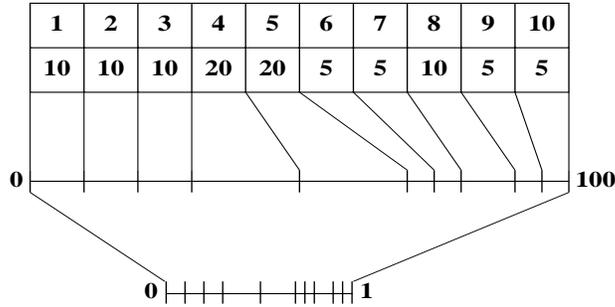


FIG. 2: Sélection de la roue de la fortune

Dans l'exemple présenté ci-dessous, la probabilité théorique de sélectionner l'individu  $\theta_5$  est de 20 pour cent<sup>9</sup>.

**Exemple :**

Indices	1	2	3	4	5	6	7	8	9	10
Pop. initiale	$\theta_1$	$\theta_2$	$\theta_3$	$\theta_4$	$\theta_5$	$\theta_6$	$\theta_7$	$\theta_8$	$\theta_9$	$\theta_{10}$
Fitness	3.2	0.5	0.2	1.5	2.5	0.3	0.2	0.4	1.5	0.3
Proba. $P_i$	0.30	0.04	0.01	0.14	0.23	0.02	0.01	0.03	0.14	0.02
Nouvelle pop.	$\theta_1$	$\theta_4$	$\theta_2$	$\theta_6$	$\theta_9$	$\theta_5$	$\theta_1$	$\theta_9$	$\theta_5$	$\theta_7$

Mais au regard de la faible dimension de cette population (10) on constate qu'il sera difficile d'obtenir cette espérance mathématique de sélection en raison du peu de tirages effectués. Un biais de sélection plus ou moins fort existe suivant la dimension de la population. Certains individus sont ainsi représentés plusieurs fois ( $\theta_5, \theta_1$ ), tandis que d'autres disparaissent ( $\theta_{10}, \theta_3, \dots$ ), d'autres enfin survivent "par chance", bien qu'ayant une adaptation faible ( $\theta_7$ ). La *roue modifiée* permet d'éviter ce genre de problèmes.

Décrivons ce principe de sélection à l'aide de l'exemple simple proposé ci-dessus ( $N = 10$ ) :

Le principe consiste à créer un sous-tableau  $T_M$  un tableau de dimension  $M$  ( $M < N$ ) tel que :

<sup>9</sup>On calcule cette probabilité théorique en calculant le rapport de la fitness de l'individu à la somme des fitness des autres individus, ici :  $\frac{2.5}{10.6}$

- Tous les individus ayant une fitness supérieure à la moyenne figurent dans  $T_M$ ,
- Chaque individu  $\theta_i$  est représenté  $N_i$  fois où  $N_i$  est la partie entière du rapport de la fitness à la moyenne des fitness  $\mu$ ,

$$N_i = Ent \left( \frac{f(\theta_i)}{\mu} \right)$$

Sur notre exemple,  $\mu = 1.06$ , ainsi l'individu aura 3 représentants dans ce tableau ( $f(\theta_1)/\mu = 3.01$ ). Pour notre exemple,  $T_M$  est de dimension  $M = 7$  :

Indices du tableau	1	2	3	4	5	6	7
Individus	$\theta_1$	$\theta_1$	$\theta_1$	$\theta_4$	$\theta_5$	$\theta_5$	$\theta_9$

La création de ce tableau est purement déterministe. L'assurance d'un nombre précis de représentants pour la génération suivante élimine le biais du principe de sélection décrit précédemment. L'ajout d'un principe aléatoire permet de ne pas éliminer complètement les mauvais individus. En effet, dans la pratique, ces individus sont nécessaires car ils peuvent occuper des positions stratégiques pour l'obtention de l'optimum. Le tableau  $T_M$  est alors étendu à  $T_N$  de dimension  $N$ , de la façon suivante :

- Pour chaque individu  $\theta_i$ , on calcule la partie fractionnaire du rapport de sa fitness à la moyenne (on obtient donc un nombre  $\alpha_i$  entre 0 et 1).
- On tire ensuite aléatoirement  $\eta$  entre 0 et 1, autant de fois qu'il manque d'individus, c'est à dire  $(N - M)$  fois .
  - Si  $\eta < \alpha_i$ , l'élément  $\theta_i$  est ajouté dans le tableau  $T_M$ .
  - Sinon, on passe à l'indice  $i + 1$ .
- Quand on arrive à l'élément d'indice  $N$  on repasse à l'élément 1.

Sur notre exemple,  $T_M$  pourrait être complété de la façon suivante :

Indices du tableau	1	2	3	4	5	6	7	8	9	10
Individus	$\theta_1$	$\theta_1$	$\theta_1$	$\theta_4$	$\theta_5$	$\theta_5$	$\theta_9$	$\theta_1$	$\theta_3$	$\theta_6$

A cette étape du processus, on dispose déjà de "bons" individus pour la nouvelle population, certains ont déjà été éliminés (comme  $\theta_2$ ,  $\theta_7$ , et  $\theta_{10}$ ). Pour assurer la non homogénéité de la population, on effectue un brassage aléatoire du tableau d'indices  $T_N$  avant de reconstituer la nouvelle population et d'élargir la recherche à l'aide des opérateurs de croisement et de mutation.

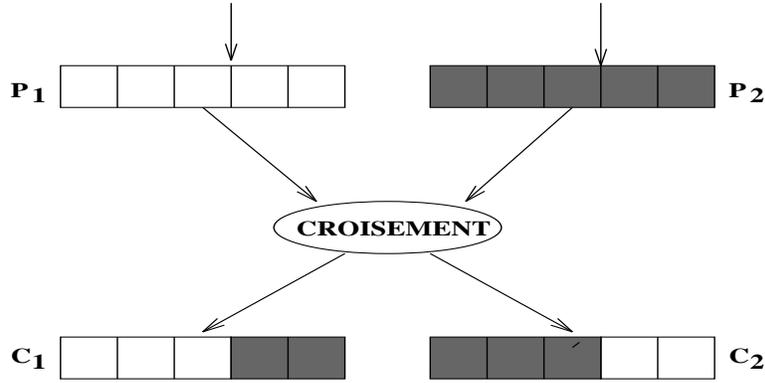


FIG. 3: Croisement chromosomique à un point

### 2.2.5 Croisement

Le croisement a pour but d'enrichir la diversité de la population en manipulant les composantes des individus (chromosomes). Classiquement, les croisements sont envisagés avec deux parents et génèrent deux enfants.

Initialement, le croisement associé au codage par chaînes de bits ou chromosomes, est le croisement à découpage de chromosomes (*slicing crossover*). Pour effectuer ce type de croisement sur des chromosomes constitués de  $M$  gènes, on tire aléatoirement une position de découpage. On échange ensuite les deux sous-chaînes terminales de chacun des deux chromosomes (les parents)  $P_1$  et  $P_2$ , ce qui produit deux nouveaux chromosomes (les enfants)  $C_1$  et  $C_2$  (voir figure 4).

On peut étendre ce principe en découpant le chromosome non pas en deux sous-chaînes mais en trois, quatre, etc.. Ce type de croisement est illustré par la figure ?? (voir Bridges et Goldberg [?]).

Le croisement à découpage de chromosomes est très rapide à mettre en oeuvre lorsqu'on travaille sur des problèmes utilisant des codages entiers (binaires ou autres).

Pour les problèmes où l'on utilise un codage réel, un croisement "barycentrique" est souvent utilisé. Deux parents  $P_1$  et  $P_2$  sont sélectionnés ; deux nouveaux points sont créés sur la droite qui les relie créant ainsi  $C_1$  et  $C_2$  de la façon suivante (voir également la figure 5) :

$$\begin{cases} C_1 = \alpha P_1 + (1 - \alpha)P_2 \\ C_2 = (1 - \alpha)P_1 + \alpha P_2 \end{cases}$$

où  $\alpha$  est un coefficient de pondération aléatoire adapté au domaine d'extension des gènes<sup>10</sup>.

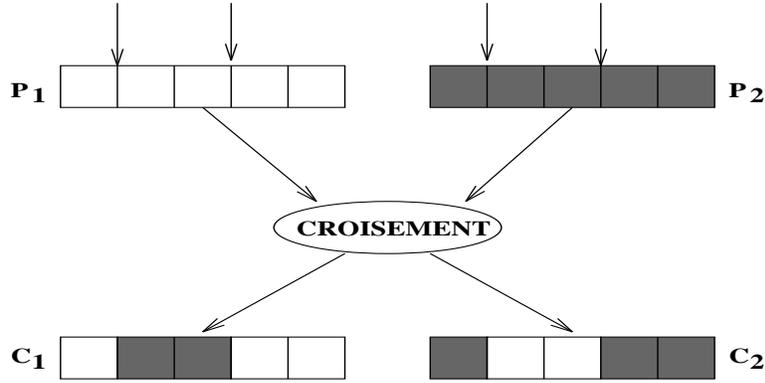


FIG. 4: Croisement chromosomique à deux points

Le croisement barycentrique permet de générer des points entre, ou à l'extérieur des deux éléments considérés. Toutefois ce type de croisement peut connaître des limitations importantes si les individus se situent sur la même droite (ou plus généralement dans le même sous espace). L'opérateur de croisement, utilisé seul, ne permet alors pas de rechercher les nouveaux éléments en dehors de cette droite.

Dans le cas particulier d'un chromosome matriciel constitué par la concaténation de vecteurs, on peut étendre ce principe de croisement aux vecteurs constituant les gènes. Deux composantes  $\vec{P}_1(i)$  et  $\vec{P}_2(i)$  sont sélectionnés dans chacun des parents à la même position  $i$ . Ils définissent deux nouveaux éléments par pondération. On crée ainsi  $\vec{C}_1(i)$  et  $\vec{C}_2(i)$  de la façon suivante :

$$\begin{cases} \vec{C}_1(i) = \alpha \vec{P}_1(i) + (1 - \alpha) \vec{P}_2(i) \\ \vec{C}_2(i) = (1 - \alpha) \vec{P}_1(i) + \alpha \vec{P}_2(i) \end{cases}$$

On peut imaginer et tester des opérateurs de croisement plus ou moins complexes sur un problème donné mais l'implémentation de ces principes est souvent liée intrinsèquement au problème.

**Remarque :**

Cette méthode de diversification des éléments rappelle la méthode du *simplexe* ou des *polytopes*. Dans cette méthode, les solutions du problèmes sont recherchée de manière itérative en considérant une population initiale<sup>11</sup>

<sup>10</sup>Il n'est pas nécessairement compris entre 0 et 1, il peut par exemple prendre des valeurs dans l'intervalle  $[-0.5, 1.5]$  afin de générer des points hors du segment, et éviter un appauvrissement de population.

<sup>11</sup>Cette population est constituée de  $d + 1$  éléments où  $d$  est la dimension de l'espace admissible.

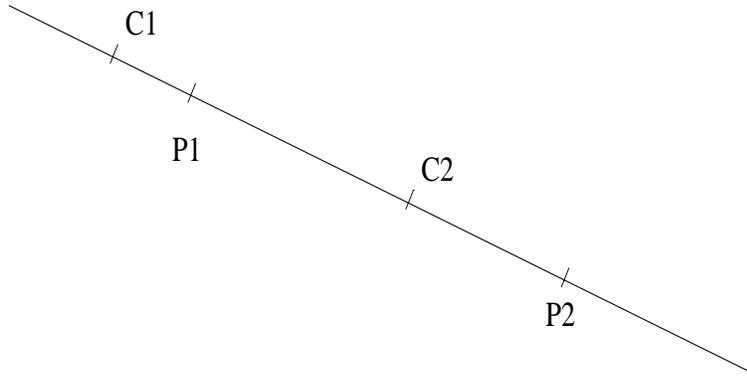


FIG. 5: Croisement barycentrique

constituant des sommets de polytopes de l'espace d'état et où les éléments successifs sont construits de manière barycentriques. Nous pouvons pousser la comparaison, puisque les éléments sont réévalués après cette diversification dans les deux méthodes. Les algorithmes génétiques offrent cependant un aspect stochastique important (les éléments sont reproduits avec la probabilité  $P_c$ ) et une efficacité supérieure.

### 2.2.6 Mutation

L'opérateur de mutation apporte aux algorithmes génétiques l'aléa nécessaire à une exploration efficace de l'espace. Cet opérateur nous garantit que l'algorithme génétique sera susceptible d'atteindre tous les points de l'espace d'état, sans pour autant les parcourir tous dans le processus de résolution. Ainsi, en toute rigueur, l'algorithme génétique peut converger sans croisement, et certaines implantations fonctionnent de cette manière (Fogel et al.[?]) et sont conformes aux résultats théoriques de R. Cerf [11], voir section 5. Les propriétés de convergence des algorithmes génétiques sont donc fortement dépendantes de cet opérateur.

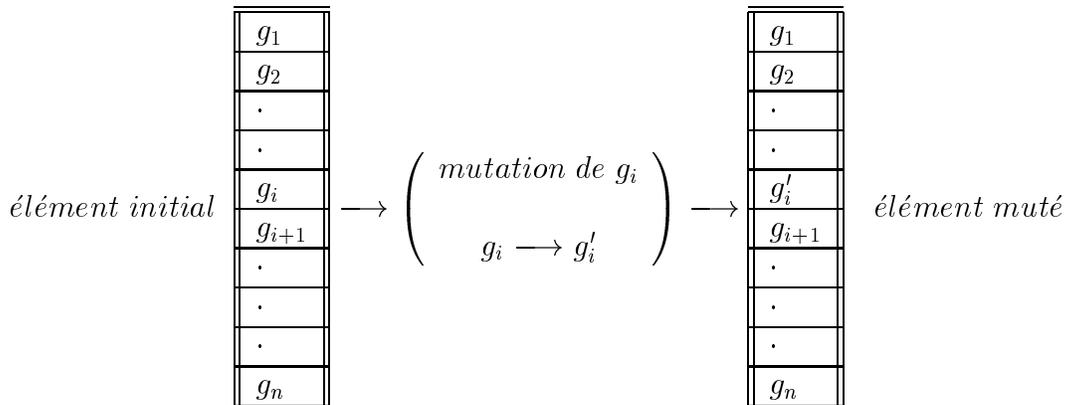
Pour bien comprendre l'utilité de l'opérateur de mutation, considérons un problème discret pour lequel les individus sont codés sous forme de chaînes chromosomiques constituées de trois valeurs entières et prenons une population de dix individus. On suppose de plus que le croisement utilisé est un croisement à découpage de chromosomes à un point peu importe lequel. Si par malchance, la population initiale ne présente pas de "7" en troisième

position, par exemple :

$\theta_1 :$	5	4	0
$\theta_2 :$	7	3	3
$\theta_3 :$	4	6	6
$\theta_4 :$	2	7	4
$\theta_5 :$	6	8	5
$\theta_6 :$	5	5	2
$\theta_7 :$	9	8	8
$\theta_8 :$	4	9	9
$\theta_9 :$	7	3	0
$\theta_{10} :$	8	2	3

et que l'optimum se trouve en "4.5.7", il est impossible d'atteindre ce point avec l'opérateur de croisement. L'opérateur de mutation permet alors de faire varier aléatoirement la valeur des composantes de ces termes, c'est à dire des gènes.

La mutation consiste généralement à tirer aléatoirement un gène dans le chromosome et à le remplacer par une valeur aléatoire (voir ci dessous). Si la notion de distance existe, cette valeur peut être choisie dans le voisinage de la valeur initiale. Dans le cas d'éléments codés en binaire, on remplacera la valeur "0" par "1" et vice-versa.



Dans les codages réels, on procède un peu de la même manière en tirant aléatoirement un élément, auquel on ajoute un bruit aléatoire en veillant à ce que l'élément résultant reste dans le domaine d'extension qui lui est propre.

Le choix pratique des probabilités  $P_m$  et  $P_c$  dépend de la complexité du problème étudié, cependant il paraît souhaitable de faire dépendre ces paramètres de la génération courante, afin d'explorer plus largement l'espace, au "début" de l'algorithme, et moins sur la "fin".

Comme nous l’avons déjà précisé, les opérateurs de croisement et de mutation ne font pas intervenir la fonction à optimiser, ce sont des opérateurs stochastiques d’exploration du domaine admissible. C’est l’opérateur de sélection qui guide la population vers les valeurs élevées de la fonction.

## 3 Améliorations classiques

### 3.1 Introduction

Les processus de sélection présentés sont très sensibles aux écarts d’adaptation entre individus et dans certains cas, un très bon individu risque d’être reproduit trop souvent et peut même provoquer l’élimination complète de ses congénères ; on obtient alors une population homogène contenant un seul type d’individu. Ainsi, dans l’exemple de la figure 6 le second mode  $M_2$  risque d’être le seul représentant pour la génération suivante et seule la mutation pourra aider à atteindre l’objectif global  $M_1$  au prix de nombreux essais successifs.

Pour éviter ce comportement, il existe d’autres modes de sélection ainsi que des principes (*scaling*, *sharing*) qui empêchent les individus “forts” d’éliminer complètement les plus “faibles”.

### 3.2 Scaling

Le *scaling* ou *mise à l’échelle*, modifie les fitness afin de réduire ou d’amplifier artificiellement les écarts entre les individus. Le processus de sélection n’opère plus sur la fitness réelle mais sur son image après *scaling*. On souhaite ainsi aplatir ou dilater la fonction d’évaluation. La perte de précision<sup>12</sup> qui s’ensuit est au prix d’une meilleure convergence.

Parmi les fonctions de *scaling*, on peut envisager le *scaling* linéaire et le *scaling* exponentiel.

---

<sup>12</sup>Une procédure d’optimisation “classique” peut être ajoutée à la fin de l’algorithme de manière à affiner le résultat.

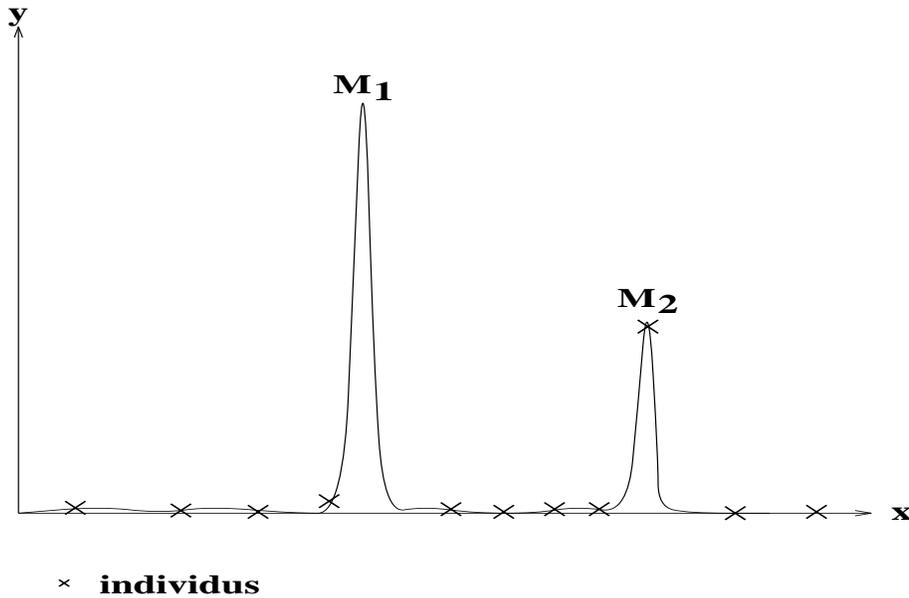


FIG. 6: Exemple où les sélections classiques risquent de ne reproduire qu'un individu

### 3.2.1 Scaling linéaire

Dans ce cas la fitness initiale  $f_r$  est redéfinie en une nouvelle fitness  $f_s$  par l'opération homothétique suivante (voir Michalewicz [21])

$$f_s = a f_r + b$$

En règle générale, le coefficient  $a$  est inférieur à un, ce qui permet de réduire les écarts de fitness et donc de favoriser l'exploration de l'espace. Par contre ce scaling est statique par rapport au numéro de génération et pénalise la fin de convergence lorsque l'on désire favoriser les modes dominants.

### 3.2.2 Scaling exponentiel

Il est défini de la façon suivante (voir figure 7) :

$$f_s = (f_r)^{n(k)}$$

où  $k$  est la génération courante.

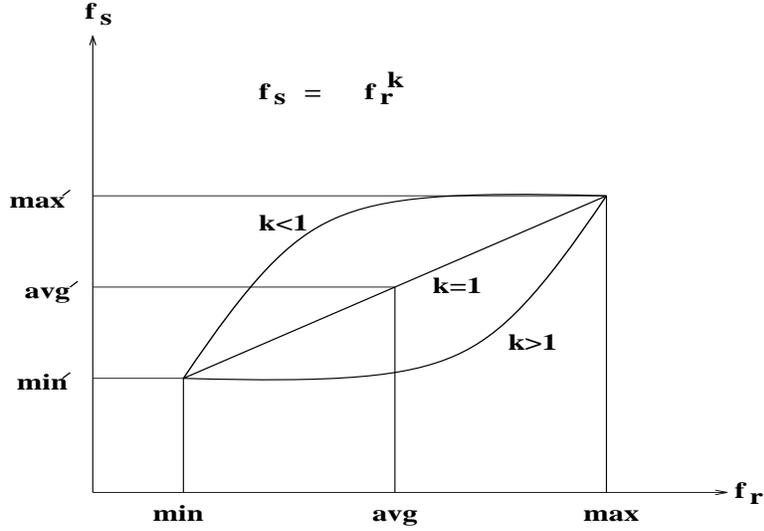


FIG. 7: Fonction de scaling exponentielle

- Pour  $n(k)$  proche de zéro, on réduit fortement les écarts de fitness ; aucun individu n'est vraiment favorisé et l'algorithme génétique se comporte comme un algorithme de recherche aléatoire et permet d'explorer l'espace.
- Pour  $n(k)$  proche de 1, le scaling est inopérant.
- Pour  $n(k) > 1$  les écarts sont exagérés et seuls les bons individus sont sélectionnés ce qui produit l'émergence des modes.

Dans la pratique, on fait généralement varier  $n(k)$  des faibles valeurs vers les fortes valeurs au cours des générations. Pour cela on utilise une fonction de type de celle représentée par la figure 8, et dont la formule est donnée par :

$$n(k) = \alpha_1 \left( \tan \left[ \left( \frac{k}{K+1} \right) \frac{\pi}{2} \right] \right)^{\alpha_2}$$

$k$  étant la génération courante,  $K$  le nombre total de générations prévues, et  $\alpha_1$  et  $\alpha_2$  sont des paramètres à choisir. L'évolution de  $n(k)$  en fonction de la génération  $k$  est donnée par la figure 8.

Ce dernier principe de scaling donne dans la pratique de meilleurs résultats que le scaling linéaire. Dans le cas des fonctions objectives possédant plusieurs modes et présentant des optimaux quasi équivalents, cette technique de scaling, en amplifiant les écarts de fitness en fin de convergence, va effectivement favoriser le mode dominant mais aussi masquer les modes

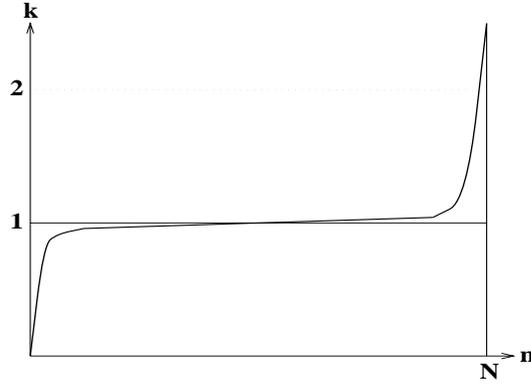


FIG. 8: Allure de l'évolution de  $n(k)$  en fonction des générations

sous-optimaux qui peuvent tout de même présenter un intérêt. Le scaling permet donc une bonne exploration de l'espace d'état mais ne favorise pas la répartition des individus sur les différents optima de la fonction objectif. Cette répartition souhaitable est obtenue en utilisant l'opérateur de partage.

### 3.3 Partage (sharing)

L'objectif du partage est de répartir sur chaque sommet de la fonction à optimiser, un nombre d'individus proportionnel à la fitness associée à ce sommet. La figure 9 présente deux exemples de répartitions de populations dans le cas d'une fonction à cinq sommets : le premier sans partage, le second avec partage.

#### 3.3.1 Principe

De la même façon que le scaling, le partage consiste à modifier la fitness utilisée par le processus de sélection. Pour éviter le rassemblement des individus autour d'un sommet dominant, le partage pénalise les fitness en fonction du taux de concentration (voir la remarque ci-dessous) de la population dans le voisinage d'un individu. Plus cet individu est entouré, plus on pénalisera sa reproduction<sup>13</sup>. Il requiert l'introduction d'une notion de distance. Dans la pratique, il faut définir une distance indiquant la dissimilarité entre deux

<sup>13</sup>Un principe d'élitisme sera toutefois appliqué afin de conserver le meilleur élément d'un groupe.

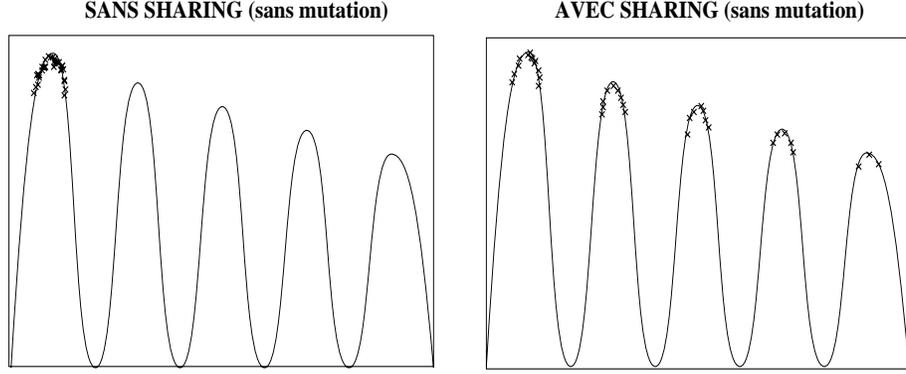


FIG. 9: Objectif du “partage”

individus. Ce qui n’est pas toujours chose facile<sup>14</sup>. Cette distance est alors utilisée pour calculer la nouvelle fitness de la façon suivante :

$$f'(\theta_i) = \frac{f(\theta_i)}{m'_i} \quad \text{où} \quad m'_i = \sum_{j=1}^N S(d(\theta_i, \theta_j)) \quad (1)$$

avec

$$S(d) = \begin{cases} 1 - \left(\frac{d}{\sigma_{share}}\right)^\alpha & \text{si } d < \sigma_{share} \\ 0 & \text{si } d > \sigma_{share} \end{cases}$$

Le paramètre  $\sigma_{share}$  permet de délimiter le voisinage d’un point et dépend du problème traité. La figure 10 donne l’allure de  $S(d)$  pour différentes valeurs de  $\alpha$ .

**Remarque :**

Ce calcul du terme pénalisant (au dénominateur) est très proche d’un calcul de densité par estimation non paramétrique. Le dénominateur  $m'_i$  intervenant en (1), jouant le rôle d’estimateur de la densité autour de l’individu  $\theta_i$ , la fonction  $S$  faisant office de noyau et  $\sigma_{share}$  de fenêtre. Il s’agit donc d’un opérateur qui pondère la fitness d’un individu par l’inverse de sa densité. On pénalise donc bien les individus se situant dans une zone “dense”.

Suivant la valeur donnée à  $\alpha$  le partage sera plus ou moins efficace. Ainsi pour  $\alpha < 1$ , on pénalise les groupes très agglomérés. A titre d’exemple, la

---

<sup>14</sup>Comment, en effet, déterminer la distance séparant deux stratégies d’enchères ? Deux équilibres de Nash dans un jeu en comportant d’autres ? Deux chemins dans le problème du voyageur de commerce ?

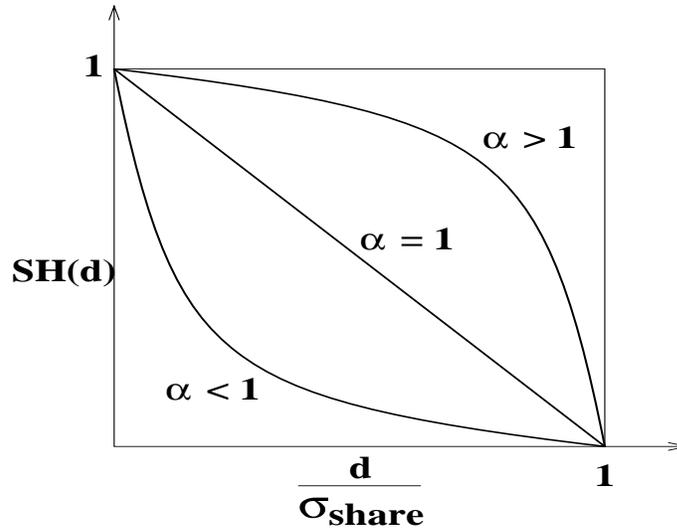


FIG. 10: Allure de  $S(\frac{d}{\sigma_{share}})$

figure ?? donne les fitness moyennes après partage pour deux répartitions possibles d’une population, dans le cas d’une fonction bi-modale (dans le cas 1, tous les points sont sur le même sommet, dans le cas 2 les points sont répartis sur les deux sommets). Le cas 2 donne une fitness moyenne, sur les individus, double de celle du cas 1, ce qui justifie l’intérêt du partage.

Dans la pratique ce type de partage donne effectivement de bons résultats mais au prix de  $N^2$  calculs de distances entre chromosomes à chaque génération pour une population de taille  $N$ . Or les algorithmes génétiques induisent une complexité en  $N$  sans partage et le fait de passer en  $N^2$  est très pénalisant.

Pour réduire ce nombre de calculs, on utilise un partage par “bouquets”<sup>15</sup>.

### 3.3.2 Partage par bouquets (Clustered sharing)

Pour effectuer ce type de partage (voir [?]), on commence par identifier les différents “bouquets” d’individus dans la population. On utilise pour cela deux paramètres  $d_{min}$  et  $d_{max}$  afin de fusionner des bouquets ou en créer de nouveaux. Initialement, chaque individu de la population est considéré comme le centre d’un bouquet. On applique alors successivement deux principes, l’un intervenant sur les bouquets, le second sur les individus eux-mêmes :

<sup>15</sup>Mot qu nous utiliserons pour traduire le terme “cluster”

FIG. 11: Influence du partage dans le cas d'une fonction bi-modale

- si deux centres sont à une distance inférieure à  $d_{min}$ , on fusionne ces derniers dans un bouquet unique dont le centre résultant est le barycentre des deux centres initiaux.
- un nouvel individu est agrégé à un bouquet si sa distance au centre le plus proche est inférieure à  $d_{max}$ .
  - Dans ce cas, on calcule de nouveau le centre du bouquet global.
  - Sinon, on crée un nouveau bouquet contenant ce seul individu. Cet individu devient alors le centre d'un nouveau bouquet réduit à lui seul.

Ce principe de fusion-agrégation permet d'obtenir un nombre de bouquets fluctuant avec la répartition des individus dans l'espace d'état. On applique ensuite le principe de partage en modifiant les fitness de la façon suivante :

$$f'_i = \frac{f_i}{m'_i}; \quad \text{ou} \quad m'_i = n_c \left( 1 - \left( \frac{d_{ic}}{2d_{max}} \right)^\alpha \right)$$

avec

- $n_c$  : nombre d'individus contenus dans le bouquet auquel appartient l'individu  $i$ .
- $\alpha$  : coefficient de sensibilité.
- $d_{ic}$  : distance entre l'individu  $i$  et le centre du bouquet  $c$ .

Ces améliorations ont été développées au sein du LOG<sup>16</sup>

**Remarque :**

La quantité  $m'_i$  représente une sorte de densité intra-bouquet, on cherche ainsi à pénaliser les bouquets trop denses ou trop importants au profit de bouquets plus petits, qui peuvent révéler des optimums locaux.

On montre que ce type de partage induit une complexité en  $O(N \log N)$  (voir e.g.[?]), pour des résultats tout à fait comparables à ceux fournis par le partage classique. Dans la pratique, on remarque que le réglage des coefficients  $d_{min}$  et  $d_{max}$  est assez délicat car l'efficacité de ces derniers dépend essentiellement de la connaissance a priori des distances entre les optimums dans l'espace d'état, distance inconnue qu'il est très difficile d'estimer.

## 3.4 Algorithmes génétiques et recuit simulé

### 3.4.1 Introduction

Les algorithmes génétiques et le recuit simulé étant deux techniques d'optimisation stochastique travaillant sur les mêmes types de problèmes, il est logique de chercher à les associer afin de tirer partie de leurs avantages respectifs. Il semble (voir Chansou [?]) que le recuit simulé converge plus vite vers la solution optimale mais ne donne qu'une solution dans le cas des problèmes à optimums multiples, ceci confirme les résultats donnés dans ??? [?]. A l'inverse, les algorithmes génétiques fournissent plusieurs solutions quasi-optimales mais au prix d'un temps de convergence plus long. Il semble alors naturel d'associer ces deux techniques afin d'améliorer la convergence des algorithmes génétiques.

Il y a eu de nombreuses tentatives de fusion entre les algorithmes génétiques et le recuit simulé, les travaux les plus intéressants étant ceux de Mahfoud et Goldberg [?].

## 3.5 Recuit simulé

---

<sup>16</sup>Laboratoire d'Optimisation Globale CENA-ENAC (Centre d'Etudes de la Navigation Aérienne, Ecole Nationale de l'Aviation Civile).

### 3.5.1 Principe

## 3.6 Croisement avec recuit

### 3.6.1 Principe du croisement avec recuit simulé

Pour appliquer ce principe de croisement, on commence par sélectionner deux parents  $P_1$  et  $P_2$  dans la population (voir figure ??). On applique ensuite l'opérateur de croisement classique qui génère deux enfants  $C_1$  et  $C_2$ . Un tournoi est alors effectué entre les parents et les enfants pour lequel les deux vainqueurs sont sélectionnés par le schéma de recuit suivant. On considère l'individu  $C_1$ . On tire ensuite aléatoirement un des deux parents, soit  $P_i$  ce parent :

- si  $C_1$  est meilleur que  $P_i$  alors  $C_1$  est sélectionné.
- sinon  $C_1$  est sélectionné avec la probabilité :

$$P = e^{-\left(\frac{|f_{C_1} - f_{P_i}|}{t_n}\right)}$$

où  $t_n$  est un coefficient décroissant en fonction de la génération courante ( $n$ ).

On fait de même pour  $C_2$  avec le parent restant et l'on détermine ainsi deux individus  $C'_1$  et  $C'_2$ .

L'évolution de la variable  $t_n$  se fait de la façon suivante. On utilise un schéma de recuit standard géométrique à un palier de basculement. Pratiquement, on calcule trois "températures" dont les valeurs dépendent de la connaissance des écarts  $min$  et  $max$  des fitness de la population initiale.

$$\begin{cases} t_s = -\frac{\Delta f_{max}}{\ln\left(\frac{1}{k-1}\right)} & k = 0.75 & \text{Température initiale} \\ t_x = -\frac{\Delta f_{max}}{\ln\left(\frac{1}{k-1}\right)} & k = 0.99 & \text{Température de basculement} \\ t_f = -\frac{\Delta f_{min}}{\ln\left(\frac{1}{k-1}\right)} & k = 0.99 & \text{Température finale} \end{cases}$$

où  $\Delta f_{min}$ ,  $\Delta f_{max}$  représentent les écarts minimum et maximum des fitness de la population initiale. Le schéma géométrique fait évoluer la température courante de la façon suivante :

$$\begin{cases} t_0 = t_s \\ t_{n+1} = \alpha_1 t_n & \text{pour } t_s > t_n > t_x; \\ t_{n+1} = \alpha_2 t_n & \text{pour } t_x > t_n > t_f; \end{cases}$$

avec  $0 < \alpha_1 < \alpha_2 < 1$ .

Pour chaque palier, on calcule le nombre d'itérations de stabilisation à l'aide des formules :

$$N_1 = \frac{\ln\left(\frac{t_x}{t_s}\right)}{\ln \alpha_1} \quad N_2 = \frac{\ln\left(\frac{t_f}{t_x}\right)}{\ln \alpha_2}$$

Ces deux formules, nous permettent de calculer le nombre total de générations pour un problème donné.

## 4 Utilisation et exemples

Comme nous l'avons déjà remarqué ces techniques permettent de fournir des éléments de population correspondant à une "zone optimale" pour un problème donné. Ces éléments, ou individus, ne donnent pas précisément *une* solution du problème mais *des* positions proches de l'optimum. C'est en sens qu'il faut interpréter les algorithmes de façon pratique. Ainsi, lorsqu'on utilise un algorithme génétique dans une optique de maximum de vraisemblance par exemple, il convient de prolonger cette procédure par un algorithme classique "grimpeur", afin d'augmenter la précision de la recherche et d'obtenir  $le(s)$  maximum(s) de manière précise (voir Goffe et al. [16], ou Dorsey et Mayer [12]).

Dans un cadre économique, il s'agira moins de déterminer le maximum précisément que d'en comprendre la signification. En effet, si l'on modélise la stratégie d'un joueur par ses actions passées dans un jeu répété, la population initiale comportera des éléments interprétables, puisque issus de comportements théoriques que l'on souhaite examiner. Il n'en sera pas de même de la population après quelques générations, les opérateurs de croisement et de mutation ayant effectué des bouleversements importants dans la structure même des individus (voir Andreoni et Miller [2], ou les vulgarisations autour du dilemme du prisonnier [?] et [?]).

Ces mêmes algorithmes génétiques peuvent également être utilisés comme aides à la décision dans une situation évoluant rapidement, on ne demande alors pas *la* meilleure solution mais une solution *raisonnable* dans un temps

limité. Cette situation, qui est à l'étude dans le cadre de procédures d'évitement d'avions (voir Durand[13]), pourrait également s'appliquer aux marchés financiers.

Nous proposons ici une revue d'ensemble des problèmes économiques et économétriques traités, en totalité ou en partie, par algorithmes génétiques.

Optimisation (Alliot et Schiex [1])

Economie : Exemple du dilemme du prisonnier d'Axelrod [6] et des enchères d'Andréoni et Miller [2]

Econométrie 11 exemples( Dorsey et Mayer [12]) +2 en projet (avec C. Bisiere et E. Malin).

Autres : Evitement des avions (Durand [13]), Othello (Alliot et Schiex [1]), Sectorisation (Delahaye [?]), etc..

## 5 Résultats théoriques

Les algorithmes génétiques ont monté leur efficacité pratique bien avant que les résultats de convergence théorique ne soient établis. Nous disposons aujourd'hui de trois approches théoriques différentes permettant de mieux comprendre le fonctionnement des algorithmes génétiques, ces trois approches donnant des résultats asymptotiques.

- La théorie des Schémas développée par Holland [18], constitue une première approche du problème. S'appliquant sur des chaînes de bits, elle étudie le comportement asymptotique de l'algorithme et l'effet des différents opérateurs sur la structure des schémas, que nous détaillerons pas ici.
- La deuxième approche découle des résultats de convergence stochastique sur des méthodes de recuit simulé développées par Laarhoven et Aarts [?]. Sous certaines hypothèses, on montre la convergence asymptotique grâce à l'opérateur de mutation. Ce résultat est d'ailleurs conforme à l'intuition, puisque seul cet opérateur permet réellement l'exploration aléatoire de l'espace.
- L'approche théorique la plus récente est proposée par Raphaël Cerf [11] et utilise une modélisation par chaîne de Markov de l'algorithme génétique. Les résultats asymptotiques sont obtenus grâce à la théorie de Freidlin et Wentzell [15].

Cette dernière approche est la plus satisfaisante tant sur le plan mathématique, que sur celui de la modélisation, les différents opérateurs étant

présentés comme “perturbant” un processus Markovien représentant la population à chaque étape. Ici encore il est démontré l’importance de l’opérateur de mutation, le croisement pouvant être totalement absent. Nous présentons ici une version simplifiée de cette théorie et les principaux résultats de convergence.

## 5.1 Modélisation de l’algorithme génétique

Les principaux résultats asymptotiques portant directement sur les algorithmes génétiques, ont été développés par R. Cerf [11] sur la base des travaux de Catoni [10] et de Trounev [24]. Ces travaux se fondent sur la théorie des petites perturbations aléatoire d’un processus dynamique de type Markovien. Plus particulièrement, la théorie de Freidlin et Wentzell [15] constitue la pierre d’angle de ces études. Nous donnons ici, quelques résultats particulièrement révélateurs de la dynamique des algorithmes génétiques, développés par Cerf. Nous les présentons simplifiés et dans un cadre restreint, laissant le lecteur intéressé se reporter à la difficile lecture des références citées ici.

Afin de préciser le cadre de cette section, nous travaillerons ici sur la base d’un codage binaire à  $P$ ,  $P$  représente le nombre d’éléments binaires (bits) utilisés pour le codage (voir section 2.2 pour plus de détails). La fonction d’évaluation,  $f$  sera donc définie sur l’espace  $E = \{0, 1\}^P$  à valeurs dans  $\mathbb{R}^+$ . Le problème est donc de localiser l’ensemble des maxima (globaux ou non) de  $f$ , ou, à défaut, de trouver rapidement et efficacement des régions de l’espace, où se situent ces maxima.

Comme nous l’avons vu l’algorithme génétique est un algorithme stochastique itératif qui opère sur des ensembles de points, et qui est bâti à l’aide de trois opérateurs : mutation, croisement et sélection, que nous présentons plus formellement à présent.

## 5.2 Description rapide de l’algorithme

Soit  $N$  la taille (fixe) de la population, notons  $X_k$  la population de la génération  $k$  : il s’agit d’une matrice  $X_k = (X_k^1, X_k^2, \dots, X_k^N)$  de  $E^N$  dont les  $N$  éléments sont des vecteurs (chromosomes) de taille  $P$  composés de 0 et de 1 (les gènes)<sup>17</sup>. Le passage de la génération  $k$  à la génération  $k + 1$ , c’est à

---

<sup>17</sup>On peut associer ces chromosomes à des “mots” composés sur l’alphabet  $\{0,1\}$ , les gènes sont alors les “lettres” 0 ou 1.

dire de  $X_k$  à  $X_{k+1}$  se décompose en trois étapes :

$$X_k \xrightarrow{\text{Mutation}} Y_k \xrightarrow{\text{Croisement}} Z_k \xrightarrow{\text{Sélection}} X_{k+1}$$

chacune de ces étapes peut être modélisée formellement.

### 5.2.1 Mutation $X_k \longrightarrow Y_k$

L'opérateur considéré ici est l'opérateur de mutation chromosomique binaire (voir section ??). Pour chaque composante de chaque élément  $X_k^i$ , une variable de Bernoulli de paramètre  $P_c$  est tirée indépendamment et suivant le résultat l'élément binaire examiné est changé ou non. S'il y a mutation les "0" sont changés en "1" et vice versa.

La probabilité  $P_c$  est la probabilité de mutation doit être préalablement choisie et est généralement "faible".

Comme nous le verrons par la suite, cet opérateur joue un rôle clé dans la convergence de l'algorithme génétique.

### 5.2.2 Croisement $Y_k \longrightarrow Z_k$

L'opérateur étudié ici est l'opérateur chromosomique à un point de découpage (slicing crossover). Ici encore, un paramètre  $P_m$  est fixé initialement, c'est la probabilité de croisement. Pour construire la population  $Z_k$ ,  $N/2$  couples sont formés à partir de la population  $Y_k$  (par exemple en appariant les individus consécutifs de  $Y_k$ , ou bien en choisissant au hasard et uniformément des individus dans  $Y_k$ ). Pour chaque couple, une variable de Bernoulli de paramètre  $P_m$  est tirée pour décider si le croisement a lieu. Si c'est le cas, un site de coupure est tiré au hasard, et les segments finaux des deux chromosomes sont échangés.

Une nouvelle paire d'individus est ainsi obtenue (identique à l'ancienne s'il n'y a pas eu de croisement) et est stockée dans la population  $Z_k$ . En général, le paramètre  $P_m$  est choisi "grand".

Remarquons que les opérateurs de mutation et de croisement ne font pas intervenir la fonction  $f$ , ce sont des opérateurs stochastiques d'exploration. C'est le troisième et dernier opérateur, la sélection, qui guide la population vers les valeurs élevées de la fonction  $f$ .

### 5.2.3 Sélection $Z_k \longrightarrow X_{k+1}$

Les  $N$  individus de la population  $X_{k+1}$  sont obtenus d'après la sélection effectuée sur les individus de  $Z_k$ . On sélectionne ainsi les "meilleurs" individus de  $Z_k$ , indépendamment à l'aide d'une distribution de probabilité qui favorise les individus de  $Z_k$  les mieux adaptés.

Le choix le plus fréquent est l'unique distribution telle que la probabilité d'un individu soit proportionnelle à son adaptation, i.e la probabilité de sélection de l'individu  $Z_k^i$  est :

$$P_i = P(Z_k^i) = \frac{f(Z_k^i)}{\sum_{j=1}^N f(Z_k^j)}$$

En tirant les individus dans la population  $Z_k$  conformément aux probabilités  $P_i$ , on constitue la nouvelle génération  $X_{k+1}$ .

## 5.3 Modélisation

La présentation rapide des opérateurs nous permet de modéliser la suite des  $(X_k)_{k \in \mathbb{N}}$  en une chaîne de Markov, d'espace d'états  $E = (\{0, 1\}^P)^N$ . L'algorithme génétique ne doit donc pas être interprété comme une procédure d'optimisation mais plutôt comme une marche aléatoire dans l'espace d'état, attirée vers les fortes valeurs de  $f$ .

La propriété première de cette formalisation est que la loi de  $X_k$  est déterminée de manière unique par :

- la loi de la génération initiale  $X_0$
- le mécanisme de transition de  $X_k$  à  $X_{k+1}$ , mécanisme scindé en les trois étapes détaillée ci-dessus.

Ce mécanisme de transition possède toutefois des propriétés essentielles qui font l'intérêt et la puissance de cette formalisation, voir Cerf ([11], chapitre 1) :

- Il est homogène (c'est à dire indépendant de la génération,  $k$ , considérée)
- Il est irréductible, (la probabilité de joindre deux points quelconques de l'espace d'état, en un nombre fini de générations est non nulle) soit :

$$\forall x, y \in E \quad \exists r \in \mathbb{N} \quad P[X_{k+r} = y \mid X_k = x] > 0$$

Le mécanisme permet donc d'explorer tout point de l'espace d'état, avec une probabilité non nulle.

– Il est apériodique, cette hypothèse n'est cependant pas fondamentale.

Ces propriétés permettent de conclure à l'ergodicité de la chaîne de Markov, et à l'existence d'un processus limite.

**Théoreme 1** *Une chaîne de Markov homogène irréductible apériodique d'espace d'états fini est ergodique et possède une unique mesure de probabilité stationnaire ou invariante.*

Cette mesure stationnaire correspond à la loi régissant l'équilibre du processus, elle est définie, pour tout  $y$ , comme :

$$\mu(y) = \lim_{k \rightarrow \infty} P[X_k = y \mid X_0 = x]$$

Nous savons également que tout élément de l'espace d'état est de probabilité non nulle pour cette mesure.

Toutefois, si ce résultat nous permet de savoir qu'il existe une dynamique de fond de l'algorithme génétique, il nous reste à en déterminer les propriétés, l'influence des opérateurs (et des paramètres associés) qui jouent un grand rôle dans le processus.

Pour cela nous introduisons les notations suivantes :

### Notations

Si  $x = (x_1, \dots, x_N)$  est un élément de  $E^N$  et  $i$  un point de  $E$ , nous notons :

$$\begin{aligned} \widehat{f}(x) &= \widehat{f}(x_1, \dots, x_N) = \max \{f(x_i) : 1 \leq i \leq N\} \\ \widehat{x} &= \left\{ x_k \in \arg \max f(x) \right\} \\ [x] &= \{x_k : 1 \leq k \leq N\} \end{aligned}$$

De manière générale, les lettres  $z, y, z, u, v..$  désignent des populations, i.e. des éléments de  $E^N$ , et les lettres  $i, j$  des points de  $E$ .

#### 5.3.1 Processus de fond ( $X_k^\infty$ )

C'est à partir de ce processus de fond qu'est reconstitué l'algorithme génétique en étudiant ses perturbations aléatoires par les différents opérateurs. Il est défini comme processus limite, lorsque les perturbations ont disparu.

C'est également une chaîne de Markov sur  $E^N$  dont le mécanisme de transition est très simple puisque correspondant à la situation limite suivante :

Les  $N$  composantes de  $X_{k+1}^\infty$  sont choisies indépendamment et suivant la loi uniforme sur l'ensemble  $\widehat{X}_k^\infty$ .

- Les individus dont l'adaptation n'est pas maximale en  $k$ , sont éliminés et n'apparaissent pas dans la génération  $k + 1$ ,
- Les individus dont l'adaptation est maximale, ont des chances de survies égales

Cette chaîne est tout d'abord piégée dans l'ensemble  $S$  des populations ayant la même adaptation (ou ensemble des population d'*équi-adaptation*),

$$S = \{x = (x_1, \dots, x_N) \in E^N \quad : \quad f(x_1) = f(x_2) = \dots = f(x_N)\}$$

Cette population représente les *attracteurs* de la chaîne (voir 5.4 plus loin), puis elle est absorbée par une population uniforme, de sorte que :

$$\forall x \in E^N \quad P[\exists x_i \in \hat{x} \quad \exists K \quad \forall k \geq K \quad X_k^\infty = x_i \mid X_0^\infty = x_{ini}] = 1$$

Lorsque la population est devenue uniforme et en l'absence ici de perturbations, il ne se passe plus rien.

Ceci peut également se traduire en définissant les populations uniformes comme les *états absorbants* de la chaîne  $X_k^\infty$ .

Nous allons maintenant étudier la situation où se processus est perturbé.

### 5.3.2 Processus perturbé ( $X_k^l$ )

La modélisation proposée par Cerf, part du processus de fond ( $X_k^\infty$ ), décrit ci-dessus, qui est perturbé aléatoirement, les perturbations sont indicées par le paramètre  $l$ . La chaîne de Markov ( $X_k^\infty$ ) devient donc une suite de chaînes de Markov ( $X_k^l$ ), dont le mécanisme de transition est donné par la succession des transformations générées par les opérateurs.

$$X_k^l \xrightarrow{\text{Mutation}} U_k^l \xrightarrow{\text{Croisement}} V_k^l \xrightarrow{\text{Sélection}} X_{k+1}^l$$

Il nous faut pour cela modéliser précisément les opérateurs.

Les mutations sont définies comme de petites perturbations aléatoires indépendante des individus, de la population  $X_k^l$ . Il est assez naturel d'introduire la probabilité  $p_l(i, j)$  de transition<sup>18</sup> de mutation entre les points  $i$  et  $j$  de  $E$ , comme un noyau Markovien  $p_l$ .

Trivialement on a  $\forall i \in E \quad \sum_{j \in E} p_l(i, j) = 1$ . Sur la chaîne  $X_k^l$ , la probabilité de transition entre les points  $x$  et  $u$  de  $E^N$  est :

$$P [U_k^l = u \mid X_k^l = x] = p_l(x_1, u_1) \cdot p_l(x_2, u_2) \cdots p_l(x_N, u_N)$$

Plus précisément et afin d'analyser la dynamique de  $(X_k^l)$  lorsque  $l$  tend vers l'infini, nous reportons ici les hypothèses sur le mode et la vitesse de convergence des probabilité de transition. Pour cela nous supposons l'existence d'un noyau irréductible,  $\alpha$ , sur  $E$ , i.e. :

$\forall i, j \in E, \exists i_0, i_1, \dots, i_r$  (c'est à dire un chemin dans  $E$ ) tels que  $i_0 = i$  et  $i_r = j$  tels que :

$$\prod_{0 \leq s \leq r-1} \alpha(i_s, i_{s+1}) > 0$$

L'hypothèse d'irréductibilité du noyau  $\alpha$  est essentielle, elle assure que tout point de l'espace est potentiellement visitable.

La vitesse de convergence du noyau  $p_l$ , est caractérisée par le réel positif  $\mathbf{a}$ , tel que  $p_l$  admette le développement suivant :

$$\forall i, j \in E \quad \forall s \quad p_l(i, j) = \begin{cases} \alpha(i, j) \cdot l^{-\mathbf{a}} + o(l^{-s}) & \text{si } i \neq j \\ 1 - \alpha(i, j) \cdot l^{-\mathbf{a}} + o(l^{-s}) & \text{si } i = j \end{cases} \quad (2)$$

La condition de positivité de  $\mathbf{a}$  nous permettent de faire disparaître les perturbations lorsque  $l$  tend vers l'infini.

$$\forall i, j \in E \quad \lim_{l \rightarrow \infty} p_l(i, j) = \delta(i, j) = \begin{cases} 0 & \text{si } i \neq j \\ 1 & \text{si } i = j \end{cases} \quad (3)$$

Ici encore l'opérateur est modélisé comme effectuant de petites perturbations aléatoires sur des couples de la population  $U_k^l$ . Ces couples sont ici

---

<sup>18</sup>C'est la probabilité  $P_l(i, j)$  pour un point  $i$  de  $E$  de se transformer par mutation en un point  $j$  de  $E$

formés par les éléments successifs de la population, les transitions sont gérées par le noyau Markovien  $q_l$  sur  $E \times E$ , cette fois, de sorte que :

$$P [V_k^l = v \mid U_k^l = u] = q_l((u_1, u_2) \cdot (u_3, u_4) \cdots (u_{N-1}, u_N))$$

Pour ce noyau  $q_l$  nous supposons l'existence d'un noyau irréductible  $\beta$  sur  $E \times E$ , la vitesse de convergence est alors paramétrée par le réel positif  $\mathbf{b}$  tel que :

$$\begin{aligned} \forall (i_1, j_1) \in E \times E \quad \forall (i_2, j_2) \in E \times E \quad \forall s \\ q_l((i_1, j_1), (i_2, j_2)) = \begin{cases} \beta((i_1, j_1), (i_2, j_2)) \cdot l^{-\mathbf{b}} + o(l^{-s}) & \text{si } (i_1, j_1) \neq (i_2, j_2) \\ 1 - \beta((i_1, j_1), (i_2, j_2)) \cdot l^{-\mathbf{b}} + o(l^{-s}) & \text{si } (i_1, j_1) = (i_2, j_2) \end{cases} \end{aligned} \quad (4)$$

L'évanouissement asymptotique des croisements est également imposée par la positivité de  $\mathbf{b}$  :

$$\forall i_1, i_2, j_1, j_2 \in E \quad \lim_{l \rightarrow \infty} q_l((i_1, i_2)(j_1, j_2)) = \delta(i_1, i_2) \cdot \delta(j_1, j_2) \quad (5)$$

C'est l'opérateur le plus compliqué et également le plus important puisqu'il permet la convergence vers les optima de  $f$ .

Il est modélisé à l'aide d'une fonction de sélection  $F_l$  dont Cerf nous donne une définition précise, pouvant être résumée par :

$$\begin{aligned} F_l \quad : \quad \{1, \dots, N\} \times (\mathbb{R}^+)^N &\longrightarrow [0, 1] \\ (i, f_1, f_2, \dots, f_N) &\longrightarrow F_l(i, f_1, f_2, \dots, f_N) \end{aligned}$$

telle que :

- i)  $F(\cdot, f_1, f_2, \dots, f_N)$  est une probabilité sur  $\{1, \dots, N\}$
- ii) Cette probabilité est indépendante de l'indexation des  $f_1, f_2, \dots, f_N$  (on peut permuer les  $f_i$ )
- iii) La probabilité favorise les éléments  $i$  associés à des valeurs élevées (i.e.)

$$\text{Si } f_1 \geq f_2 \geq \dots \geq f_N \quad \text{Alors}$$

$$F_l(1, f_1, f_2, \dots, f_N) \geq F_l(2, f_1, f_2, \dots, f_N) \geq \dots \geq F_l(N, f_1, f_2, \dots, f_N)$$

Cet outil, dont on voit qu'il modélise les probabilité de sélection définies section 6, nous permet d'écrire la probabilité de transition correspondant à la dernière étape.

$$P [X_{k+1}^l = x \mid V_k^l = v] = \prod_{r=1}^N \Upsilon_l(x_r, v_r)$$

ceci signifie que la probabilité de transition est le produit des probabilités sur chacune des  $N$  composantes de  $E$ .

La probabilité  $\Upsilon_l$  entre deux composantes  $(x_r, v_r)$  est donnée par :

$$\Upsilon_l(x_r, v_r) = \sum_{k : v_k = x_k} F_l(k, f(v_1), f(v_2), \dots, f(v_N))$$

De même que pour les autres opérateurs, la fonction de sélection doit être choisie et sa vitesse de convergence caractérisée :

$$F_l(i, f_1, f_2, \dots, f_N) = \frac{\exp(\mathbf{c} \cdot f_i \cdot \ln(l))}{\sum_{r=1}^N \exp(\mathbf{c} \cdot f_r \cdot \ln(l))} \quad (6)$$

Ce choix correspond bien à une probabilité de sélection avantageant les fortes adaptation au détriment des faibles, le réel positif  $c$  indexant cette fonction.

Le mécanisme de sélection opérant sur le processus de fond  $(X_k^\infty)$ , correspond à la fonction de sélection  $F_\infty$  définie par :

$$F_\infty(k, f(x_1), f(x_2), \dots, f(x_N)) = \frac{\mathbf{1}_{\hat{x}}(x_k)}{\text{card}(\hat{x})}$$

C'est à dire, la loi uniforme sur l'ensemble  $\hat{x} = \{x_k \in \arg \max f(x)\}$

La suite  $(F_l)_{l \in \mathbb{N}}$  des fonctions de sélection tend vers cette loi uniforme

$$\forall x \in E^N \quad \forall k$$

$$\lim_{l \rightarrow \infty} F_l(k, f(x_1), f(x_2), \dots, f(x_N)) = F_\infty(k, f(x_1), f(x_2), \dots, f(x_N)) \quad (7)$$

Les conditions 3, 5, et 7 nous permettent d'assurer que le mécanisme de transition de la chaîne  $(X_k^l)$  converge vers celui du processus de fond  $(X_k^\infty)$  :

$$\forall y, z \in E^N \quad \lim_{l \rightarrow \infty} P [X_{k+1}^l = z \mid X_k^l = y] = P [X_{k+1}^\infty = z \mid X_k^\infty = y]$$

C'est également en ce sens que l'on interprète la chaîne  $(X_k^l)$  comme une perturbation de la chaîne  $(X_k^\infty)$ .

Les vitesses de convergence intervenant dans chacun des opérateurs jouent un rôle important. La formulation proposée en (2), (4) et (6), permet un ajustement équitable de ces vitesses (elles sont logarithmiquement du ordre) de sorte qu’aucun opérateur ne domine les autres dans la dynamique. lorsque  $l$  tend vers l’infini, les conditions (3), (5), et (7) nous permettent d’assurer que le mécanisme de transition de la chaîne  $(X_k^l)$  converge vers celui du processus de fond  $(X_k^\infty)$ , et on a<sup>19</sup> :

$$\forall y, z \in E^N \quad \lim_{l \rightarrow \infty} P [X_{k+1}^l = z \mid X_k^l = y] = P [X_{k+1}^\infty = z \mid X_k^\infty = y]$$

La chaîne  $(X_k^l)$  se comporte alors comme le ferait  $(X_k^\infty)$ . La théorie de Freidlin-Wentzell nous donne les outils pour simplifier l’étude de ces processus à temps continu.

## 5.4 La théorie de Freidlin et Wentzell

### 5.4.1 Principe

Soit le système différentiel de  $\mathbb{R}^N$  satisfaisant les équations déterministes :

$$\begin{cases} dx_t = b(x_t) dt \\ x_0 = x_{ini} \end{cases} \quad (8)$$

Sous de “bonnes” hypothèses, il existe une solution (trajectoire) unique,  $x(t)$  à l’équation (8) et à la condition initiale associée. l’une des préoccupations immédiates est de savoir si cette solution va, ou non, tendre vers un équilibre (qui n’est pas forcément unique). Et si oui, quel en est l’ensemble de stabilité. L’équilibre est défini comme une fonction constante  $x^*$  telle que  $x^* = \lim_{t \rightarrow \infty} x_t$ , et l’ensemble de stabilité comme l’ensemble  $K(x^*)$  des points de départ qui mènent à cet équilibre<sup>20</sup>. On peut élargir cette notion, d’équilibre et de stabilité, par celles, très proches, d’attracteur et de bassin d’attraction.

<sup>19</sup>C’est également en ce sens que l’on interprète la chaîne  $(X_k^l)$  comme une “perturbation” de la chaîne  $(X_k^\infty)$ .

<sup>20</sup>L’ensemble de stabilité de l’équilibre  $x^*$  est :

$$K(x^*) = \left\{ x_{ini} \in \mathbb{R}^N, \text{ t.q. pour } x_t \text{ solution de 8 ; } \lim_{t \rightarrow \infty} x_t = x^* \right\}$$

Pour chaque équilibre on définit ainsi son ensemble de stabilité. Cet équilibre est stable s’il contient un voisinage de l’équilibre, et instable s’il existe des points de départ infiniment proche de l’équilibre qui ne mènent pas à celui-ci.

Un attracteur du système est le voisinage compact  $K_i$  d'un point visité une infinité de fois, et le bassin d'attraction l'ensemble des points de départ qui mènent à cet attracteur. Nous supposons que  $\mathbb{R}^d$  possède un nombre fini d'attracteurs  $K_1, \dots, K_r$ .

La théorie de Freidlin-Wentzell étudie la perturbation du système 8, par des perturbation Browniennes, d'intensité  $\varepsilon$ . Le système déterministe 8 devient alors un système différentiel stochastique.

$$\begin{cases} dX_t^\varepsilon = b(X_t^\varepsilon) dt + \varepsilon d\omega_t \\ X_0^\varepsilon = x_{ini} \end{cases} \quad (9)$$

Le processus  $(X_t^\varepsilon)_{t \in \mathbb{R}^+}$  est maintenant un processus stochastique perturbé par le mouvement brownien  $(\omega_t)_{t \in \mathbb{R}^+}$  et dépendant de  $\varepsilon$ . La situation change alors puisque les perturbations brownienne permettent au processus de s'échapper de n'importe quel bassin d'attraction, et en fait le processus les visite tous.

De plus, le processus est ergodique et admet une unique mesure de probabilité invariante, i.e.

$$\forall \mathcal{B} \text{ borélien de } \mathbb{R}^d \quad \lim_{t \rightarrow \infty} P[X_t^\varepsilon \in \mathcal{B} \mid X_0^\varepsilon = x_{ini}] = \mu^\varepsilon(\mathcal{B})$$

existe est la probabilité de présence du processus dans le Borélien  $\mathcal{B}$ , lorsque le système a atteint son état d'équilibre. Cette probabilité  $\mu^\varepsilon$  est invariante avec le point de départ  $x_{ini}$ .

Lorsque les perturbation cessent, le processus se comporte comme dans 8 et reste presque sûrement au voisinage  $V(K_1 \cup \dots \cup K_r)$  des attracteur, tandis que la probabilité de présence dans n'importe quel Borélien  $\mathcal{A}$  disjoint de  $K_1 \cup \dots \cup K_r$  disparaît.

$$\lim_{\varepsilon \rightarrow 0} \mu^\varepsilon(V(K_1 \cup \dots \cup K_r)) = 1$$

$$\lim_{\varepsilon \rightarrow 0} \mu^\varepsilon(\mathcal{A}) = 0$$

Le résultat principal de Freidlin et Wentzell repose sur l'équivalence du processus  $(X_t^\varepsilon)_{t \in \mathbb{R}^+}$  à temps continu et espace d'état  $\mathbb{R}^d$  et du processus  $(Z_n^\varepsilon)_{n \in \mathbb{N}}$  à temps discret et espace d'état fini  $\{1, \dots, r\}$  décrivant les visites au  $n$ ième attracteur.

La construction précise de  $(Z_n^\varepsilon)_{n \in \mathbb{N}}$ , n'est pas reportée ici mais nous en donnons un aperçu afin de mieux comprendre ce dernier processus.

- Si  $x_{ini}$  est "proche" de l'attracteur  $K_h$  alors  $Z_0^\varepsilon = h \in \{1, \dots, r\}$

- puis le processus, sous l’influence de  $(\omega_t)$ , est attiré par  $K_s$  et  $Z_1^\varepsilon = s$
- etc..

La chaîne de Markov<sup>21</sup> ainsi créée a pour espace d’états  $\{1, \dots, r\}$ , est irréductible, et possède une unique mesure de probabilité invariante  $\nu^\varepsilon$ .

**Théoreme 2** *L’étude du comportement asymptotique de la mesure  $\mu^\varepsilon$  est “équivalente” à l’étude du comportement asymptotique de la mesure  $\nu^\varepsilon$*

Nous passons sous silence l’étude des probabilité de transition  $P[Z_n^\varepsilon = i \mid Z_n^\varepsilon = j]$  de la chaîne  $(Z_n^\varepsilon)_{n \in \mathbb{N}}$  qui s’écrivent comme des intégrales sur l’ensemble des fonctions  $\phi$  qui lient les attracteurs  $K_i$  et  $K_j$ , laissant le lecteur intéressé se reporter à la lecture de Freidlin et Wentzell [15], ou de Cerf [11].

Notons toutefois que ces probabilité de transition s’écrivent :

$$P[Z_n^\varepsilon = i \mid Z_n^\varepsilon = j] \sim \exp -\frac{V(i, j)}{\varepsilon^2}$$

où  $V(i, j) = \inf\{V(\phi), \phi(\cdot)$  continue  $[0, 1] \mapsto \mathbb{R}^d, \phi(0) \in K_i, \phi(1) \in K_j\}$  et  $V(\phi) = \int_0^1 \left| \dot{\phi}(t) - b(\phi(t)) \right|^2 dt$ . est une constant associée à  $\phi$  et caractérisant sa vitesse de convergence.

La quantité  $V(i, j)$  ou *coût de communication*, mesure le coût de passage de l’attracteur  $K_i$  à l’attracteur  $K_j$ .

Les intensités de transitions de la chaîne  $(Z_n^\varepsilon)_{n \in \mathbb{N}}$ , nous ouvrent la voie pour déterminer la mesure invariante  $\nu^\varepsilon$ .

#### 5.4.2 Mesure invariante $\nu^\varepsilon$

Les outils qui permettent de déterminer cette mesure invariante ont été développés, une nouvelle fois par Freidlin et Wentzell, nous aurons besoins de certains d’entre eux.

**Définition :** Soit  $i$  un élément de  $\{1, \dots, r\}$ .

Un  $i$ -*graphe* sur  $\{1, \dots, r\}$  est un graphe  $g$  sur  $\{1, \dots, r\}$  possédant les propriétés suivantes :

---

<sup>21</sup>La nature Markovienne de  $\omega_t$ , nous permet de montrer qu’il s’agit bien là d’une chaîne de Markov.

- $\forall j \neq i$ , le graphe  $g$  contient une unique flèche issue de  $j$
- Il existe un chemin dans  $g$  qui mène de  $j$  à  $i$
- $g$  ne contient pas de flèche issue de  $i$

Il s'agit donc d'un graphe sans cycles formé de chemins qui aboutissent en  $i$ . On note  $G(i)$  l'ensemble des  $i$ -graphes sur  $\{1, \dots, r\}$ .

**Définition :** La fonction d'énergie virtuelle  $W$  est la fonction de  $\{1, \dots, r\}$  dans  $\mathbb{R}^+$  définie par :

$$\forall i \in \{1, \dots, r\} \quad W(i) = \min_{g \in G(i)} \sum_{(\alpha \rightarrow \beta) \in g} V(\alpha, \beta)$$

a cette fonction est associé l'ensemble  $W^*$  des minima globaux de  $W$ .

Finalement, la mesure invariante  $\nu^\varepsilon$  est caractérisée par :

$$\forall i \in \{1, \dots, r\} \quad \nu^\varepsilon(i) \sim \exp - \frac{W(i) - W(W^*)}{\varepsilon^2}$$

où  $W(W^*) = \min \{W(i) : i \in \{1, \dots, r\}\}$ .

Le comportement asymptotique de  $\nu^\varepsilon$  (et par la même occasion de  $\mu^\varepsilon$ ) est donc connu : la mesure  $\nu^\varepsilon$  se concentre sur les attracteurs dont l'indice est dans  $W^*$  et décroît vers zéro à la vitesse  $\exp - \frac{Cste}{\varepsilon^2}$  pour les autres attracteurs. Il existe donc un sous-ensemble de  $W^*$  de l'ensemble des attracteurs sur lequel se concentre la mesure invariante du processus.

$$\lim_{\varepsilon \rightarrow 0} \lim_{t \rightarrow \infty} P \left[ X_t^\varepsilon \in V \left( \bigcup_{i \in W^*} K_i \right) \mid X_0^\varepsilon = x_{ini} \right] = 1$$

La dynamique du processus est donc caractérisable.

### 5.4.3 Dynamique du processus.

Dans sa thèse, Cerf nous donne une très claire interprétation de la hiérarchie des cycles qui caractérisent la dynamique du processus. Supposons que le processus soit initialement dans le bassin d'attraction de  $K_1$ . Il quitte  $K_1$  au bout d'un temps fini. Parmi toutes les trajectoires de sortie, il en existe une plus "probable" que les autres, qui l'amène vers un nouvel attracteur ; par exemple  $K_2$  puis, bientôt  $K_3$ . L'ensemble des attracteurs étant par hypothèse fini, le processus finit par revisiter un attracteur formant un cycle d'ordre 1

sur lequel le processus tourne longtemps, très longtemps. Englobons maintenant ces trois attracteurs dans une boîte. Comme toujours, les perturbations browniennes finissent par pousser le processus hors de cette boîte, et ici encore, il existe une trajectoire de sortie canonique qui fait tomber le processus dans un nouveau bassin d'attraction, ou plus généralement, dans un autre cycle d'ordre 1.

Les cycles d'ordre 1 sont aussi en nombre fini, et le processus finit par revisiter un cycle d'ordre 1 : un cycle d'ordre 2 est alors formé, dans lequel le processus reste piégé très très longtemps. En continuant de la sorte, il est possible de construire toute une hiérarchie de cycles qui épuise l'ensemble des attracteurs et fournit une image très précise de la dynamique asymptotique du processus. A chaque transition entre cycles est associée une constante qui caractérise la difficulté de la transition.

Enfin, lorsque  $\varepsilon$  décroît avec le temps, i.e.  $\varepsilon = \varepsilon(t)$  est une fonction de  $t$  qui tend en décroissant vers 0, nous obtenons un processus de Markov inhomogène (le mécanisme de transition dépend du temps).

- Si  $\varepsilon(t)$  décroît très lentement, de sorte qu'à chaque instant la loi de  $X_t$  soit proche de l'état d'équilibre associé au niveau de perturbation  $\varepsilon(t)$ , la situation ne change pas fondamentalement. La loi limite correspond à la limite de la suite des lois d'équilibre.
- Si au contraire  $\varepsilon(t)$  décroît très rapidement, le processus risque de rester piégé dans certains sous-ensembles d'attracteurs : plus précisément, dans la hiérarchie des cycles, certaines transitions ne pourront être effectuées qu'un nombre fini de fois, alors que d'autres, plus "faciles", seront réalisées une infinité de fois avec probabilité 1. La loi limite dépend alors fortement du point de départ.

La hiérarchie des cycles permet ainsi de décrire les dynamiques possibles de  $(X_t)$  en fonction de la vitesse de décroissance de  $\varepsilon(t)$ .

## 5.5 Résultats de convergence

Lorsque  $l$  croit vers l'infini, les perturbations affectant le processus  $(X_k^l)$  diminuent de sorte que cette chaîne se comporte, presque sûrement, comme la chaîne  $(X_k^\infty)$ . Plus précisément, nous savons que les attracteurs de la chaîne  $(X_k^\infty)$  sont les population d'équi-adaptation  $S$  et les populations uniformes (attracteurs stables). La chaîne  $(X_k^l)$  va donc être attirée par ses attracteurs, en commençant par l'ensemble  $S$ .

La théorie de Freidlin et Wentzell nous permet de reporter cette étude sur celle de la chaîne des  $(Z_k^l)$  des visites successives de  $(X_k^l)$  à l'ensemble  $S$ . Nous poserons donc  $Z_k^l = X_{T_k}^l$  où  $T_k$  est l'instant de la  $k$ ième visite de  $(X_k^l)$  dans  $S$ .

Les probabilité de transition de la chaîne  $(Z_k^l)$ , sont estimées à l'aide des opérateurs définis en 5.3 et selon le schéma développé ci-dessus. Les fonctions de coût de communication  $V(i, j)$  et d'énergie virtuelle  $W$  sont définies et estimées.

Nous savons alors que la suite des mesures stationnaires de la chaîne  $(X_k^l)$  se concentre sur l'ensemble  $W^*$  des minima de  $W$  :

$$\forall x_{ini} \in E^N \quad \lim_{l \rightarrow \infty} \lim_{k \rightarrow \infty} P [X_k^l \in W^* \mid X_0^l = x_{ini}] = 1$$

L'un des principaux résultats indique qu'il existe une taille de la population de  $(X_k^l)$ , (taille critique) telle que les maxima de  $f$  soient atteints asymptotiquement avec probabilité 1.

### 5.5.1 Taille critique

Supposons fixés l'espace d'état  $E$ , la fonction d'adaptation  $f$ , les noyaux de transition de mutation  $\alpha$  et de croisement  $\beta$ , ainsi que les constantes positives gérant les trois opérateurs  $\mathbf{a}$ ,  $\mathbf{b}$ , et  $\mathbf{c}$ .

#### **Théoreme 3** (Cerf 1993)

*Il existe une valeur critique  $N^*$ , telle que lorsque la taille de la population de l'algorithme génétique dépasse  $N^*$ , l'ensemble  $f^*$  des maxima globaux de  $f$ , contient l'ensemble  $W^*$ .*

*Cette taille critique  $N^*$ , dépend fortement de l'espace d'état  $E$ , de la fonction d'adaptation  $f$ , des noyaux  $\alpha$  et de croisement  $\beta$ , ainsi que des paramètres  $\mathbf{a}$ ,  $\mathbf{b}$ , et  $\mathbf{c}$ .*

Une borne grossière, mais lisible de  $N^*$  est :

$$N^* \leq \frac{\mathbf{a}R + \mathbf{c}(R - 1)\Delta}{\min(\mathbf{a}, \frac{\mathbf{b}}{2}, \mathbf{c}\delta)}$$

où :

- $R$  est le nombre minimal de transition permettant de joindre deux points arbitraires de  $E$  par mutation

–  $\Delta$  et  $\delta$  sont des paramètres d'échelle :

- $\Delta = \max \{|f(i) - f(j)| : i, j \in E\}$  paramètre mesurant les écarts maximaux de  $f$
- $\delta = \min \{|f(i) - f(j)| : i, j \in E, f(i) \neq f(j)\}$  mesurant les écarts minimaux.

Il est intéressant de relever que le résultat est obtenu sans faire intervenir l'opérateur de croisement, qui n'est donc pas indispensable. L'exploration par mutation et le guide de la force de sélection suffisent à assurer la convergence vers  $f^*$ .

Ce premier résultat nous indique que dès que  $N \geq N^*$ , la suite des mesures stationnaires de la chaîne  $(X_k^l)$  se concentre asymptotiquement sur  $f^*$  lorsque  $l$  tend vers l'infini. L'étape suivante consiste donc à faire évoluer  $l$ , et donc l'intensité des perturbations, en fonction de la génération pour obtenir un algorithme génétique correspondant à ceux présentés dans la section 2. Nous obtenons alors une chaîne de Markov inhomogène  $(X_k^{l(k)})$  dont le mécanisme de transition dépend alors de la génération  $k$ .

### 5.5.2 Vitesse de convergence

Le principal problème est de savoir si cette chaîne inhomogène peut avoir un comportement proche de celui de la chaîne homogène  $(X_k^l)$ , et si oui, sous quelles conditions. La vitesse de croissance de  $l(k)$  vers l'infini, est bien évidemment, au centre du débat.

- Si  $l(k)$  croît “lentement”, alors la loi de  $X_k$  sera proche de la loi stationnaire  $\mu^{\varepsilon(l(n))}$  de niveau de perturbation  $\varepsilon(l(n))$  associé à  $l(n)$ .
- Si  $l(k)$  croît “rapidement”, alors  $X_k$  risque de rester piégé dans des bassins d'attraction ne correspondant pas aux maxima de  $f$ , l'intensité des perturbation devenant trop faible pour pouvoir s'en échapper.

La vitesse recherchée se situe entre ces deux extrêmes, permettant à  $X_k$  de s'échapper des “mauvais” bassins d'attraction (ne correspondant pas à des maxima de  $f$ ) et de rester piégé dans le “bon” (celui des points de  $f^*$ ).

La vitesse de convergence de la suite  $l(k)$  est caractérisée par l'*exposant de convergence*<sup>22</sup>,  $\lambda$ .

---

<sup>22</sup>Egalement appelé *rayon de convergence*.

**Définition :**

L'exposant de convergence  $\lambda$  de la suite  $l(k)$  est l'unique réel  $\lambda$  tel que :

$$\begin{aligned} \sum_{k \in \mathbb{N}} l(k)^{-\theta} &\rightarrow \text{converge pour } \theta > \lambda \\ &\rightarrow \text{diverge pour } \theta < \lambda \end{aligned}$$

Deux conditions pour la colonisation de  $f^*$  sont également données par Cerf, l'une nécessaire, l'autre suffisante.

**Théorème 4** *Condition nécessaire pour la colonisation de  $f^*$* 

Pour que :

$$\forall x_{ini} \in E^N \quad P [\exists K \quad \forall k \geq K \quad [X_{T_k}] \subset f^* \mid X_0 = x_{ini}] = 1$$

*c'est à dire, pour que la chaîne  $Z_k = X_{T_k}$  des visites successives des attracteurs soit piégée dans  $f^*$  après un nombre fini  $K$  de transitions,*

*il est nécessaire que l'exposant de convergence  $\lambda$  de la suite  $l(k)$  appartienne à l'intervalle  $] \phi, \psi [$ .*

*Les constantes  $\phi$  et  $\psi$  sont des caractéristiques du problème, l'intervalle  $] \phi, \psi [$  est alors non vide pour  $N$  assez grand.*

**Théorème 5** *Condition suffisante pour la colonisation de  $f^*$* 

*Il existe deux constantes  $\eta$  et  $\rho$  tel que si l'exposant de convergence  $\lambda$  de la suite  $l(k)$  appartient à l'intervalle  $] \eta, \rho [$ , alors :*

$$\forall x_{ini} \in E^N \quad P [\exists K \quad \forall k \geq K \quad [X_{T_k}] \subset f^*, \widehat{X}_k \subset f^* \mid X_0 = x_{ini}] = 1$$

*ce qui signifie qu'après un nombre fini de transitions, nous avons presque sûrement, la situation suivante :*

- la chaîne  $X_{T_k}$  est piégée dans  $f^*$ ,
- la population  $X_k$  contient toujours un ou des individus appartenant à  $f^*$ .

**5.5.3 En guise de conclusion**

D'autres résultats existent, tant dans le travail de Cerf, que dans la littérature citée dans cette section. Ils demandent cependant un investissement

supplémentaire dans la compréhension des outils développés. Le but de cette section était de convaincre le lecteur que l'étude théorique des algorithmes génétiques donne (déjà) de substantiels résultats. De nombreuses interrogations demeurent cependant concernant les relations réelles entre les différents paramètres caractérisant l'algorithme génétique et les choix pratiques de ceux-ci. Dans ce domaine, la pratique devance encore la théorie, même si les mécanismes commencent à être plus clairs. Il reste également à étudier les nombreux raffinements présentés dans la section 2, et aujourd'hui en application. Comment incorporer les opérateurs de partage, et leur implémentation par bouquets ? Comment trouver de nouveaux algorithmes génétiques encore plus efficaces ? Autant de questions qui trouveront sans doute des réponses dans les travaux futurs.

L'exemple de l'aspirine nous vient à l'esprit, ce n'est que dans les années 70 que l'on comprit réellement les mécanismes de son action, cela ne l'a pas empêché d'agir depuis sa découverte en 18??.

## 6 Perspectives

## 7 Appendice : Un exemple élémentaire.

Soit le problème de maximisation suivant :

$$\begin{cases} \max f(x) = 4x(1-x) \\ x \in [0, 1] \end{cases}$$

La fonction  $f(x)$  admet un maximum unique en  $x = 0,5$  pour lequel  $f(x)$  vaut 1, comme le montre la représentation graphique ci-dessous.

### Codage

Afin de bien visualiser les propriétés des opérateurs nous décidons de traiter ce problème en codant les éléments de  $[0, 1]$  en chaînes de bits de longueur 8. Par exemple, 10111010 constituera un élément de la population.

### Population initiale (Génération zéro)

L'algorithme génétique consiste tout d'abord à tirer une population initiale de  $N = 4$  éléments,  $(\theta_i)_{i=1,\dots,4}$  donnés dans le tableau ci dessous, nous évaluons par la même occasion leur adaptation, c'est à dire ici  $f(\theta_i)$ .

Eléments	Eléments Codés	Adaptation : $f(\theta_i)$
$\theta_1$	10111010	0,794678
$\theta_2$	11011110	0,460693
$\theta_3$	00011010	0,364990
$\theta_4$	01101100	0,975586

(10)

Il va s'agir maintenant de sélectionner les éléments en fonction de leur adaptation. On le voit ici, les éléments  $\theta_4$  et  $\theta_1$  sont les meilleurs.

### Sélection

Pour sélectionner ces candidats à reproduction, nous allons utiliser la sélection de la roue de la fortune et attribuer à chacun une probabilité de reproduction égale à :

$$P_i = \frac{f(\theta_i)}{\sum_{j=1}^4 f(\theta_j)}$$

et donc le tableau 10 s'élargit de cette probabilité de reproduction de chaque élément.

Eléments	Eléments Codés	$f(\theta_i)$	$P_i$
$\theta_1$	10111010	0,794678	$0,794/2,59 = 0,31$
$\theta_2$	11011110	0,460693	$0,460/2,59 = 0,18$
$\theta_3$	00011010	0,364990	$0,364/2,59 = 0,14$
$\theta_4$	01101100	0,975586	$0,975/2,59 = 0,37$
<b>Cumul</b>		<b>2,593</b>	

Un problème pratique se pose : Comment "tirer" 4 nombres parmi 4 avec remplacement en affectant à chacun cette probabilité ?

Il existe pour cela une méthode simple et rapide, il suffit d'attribuer un segment de taille  $P_i$  à l'individu  $\theta_i$  et de reporter ces segments bouts-à-bouts dans l'intervalle<sup>23</sup>  $[0, 1]$ . Les individus sont identifiés par un segment particulier de longueur  $P_i$ .

Sur l'exemple cela donne :

$\theta_1$  est ainsi caractérisé par l'intervalle  $[0, 0.31]$  de longueur 0,31

$\theta_2$  est caractérisé par l'intervalle  $[0.31, 0.49]$  de longueur 0,18

$\theta_3$  par l'intervalle  $[0.49, 0.63]$  de longueur 0,14

et  $\theta_4$  par  $[0.63, 1]$  de longueur 0,37.

On tire ensuite  $\eta$  uniformément dans  $[0, 1]$  et l'on reproduit 4 fois ce tirage, on détermine ainsi 4 nouveaux éléments grâce à ce tirage. Ici le tirage de  $\eta$  donne ; 0.47 ( $\theta_2$  est sélectionné), 0.89 ( $\theta_4$ ), 0.18 ( $\theta_1$ ) et 0.75 ( $\theta_4$  de nouveau). A ce stade  $\theta_3$  est éliminé de la population tandis que  $\theta_4$  est reproduit deux fois.

Les opérateurs de croisement et de mutation s'appliqueront donc sur la nouvelle population constituée de ( $\theta_2, \theta_4, \theta_1$  et  $\theta_4$ ) renommés  $\beta_1, \beta_2, \beta_3$  et  $\beta_4$ .

Eléments	Eléments Sélectionnés	Renommés
$\theta_1$	$\theta_2 = 10111010$	$\beta_1$
$\theta_2$	$\theta_4 = 01101100$	$\beta_2$
$\theta_3$	$\theta_1 = 01101100$	$\beta_3$
$\theta_4$	$\theta_4 = 01101100$	$\beta_4$

## Croisement

<sup>23</sup>La somme des  $P_i$  vaut, en effet, 1.

La probabilité de croisement  $P_c$  est ici fixée à 50% (on ne peut faire moins étant donné la taille de la population considérée), cela signifie que l'on va tirer un couple au hasard et lui appliquer le croisement chromosomique à un point.  $(\beta_1, \beta_3)$  constitue le couple destiné à être transformé, il faut encore déterminer la position du croisement dans les composantes (gènes) de ces éléments. Cette position peut être elle aussi tirée au hasard ou choisie arbitrairement. Nous décidons d'effectuer les croisements sur le milieu afin de rendre l'exemple plus parlant.

Les parties terminales des individus  $\beta_1$  et  $\beta_3$  sont donc échangées, comme suit :

$$\begin{cases} \beta_1 = 1011\mathbf{1010} \rightsquigarrow 1011 \mathbf{1100} = \lambda_1 \\ \beta_3 = 0110\mathbf{1100} \rightsquigarrow 0110 \mathbf{1010} = \lambda_3 \end{cases}$$

engendrant ainsi deux nouveaux individus (les enfants)  $\lambda_1$  et  $\lambda_3$ .

On peut, sur la base de ces nouveaux individus muter aléatoirement l'une des composantes de l'un des individus constituant la nouvelle population ( $\lambda_1, \beta_2, \lambda_3$  et  $\beta_4$ ).

### Mutation

La probabilité de mutation est ici de  $P_c = 0.25$ , de sorte qu'un individu sur les quatre sera choisi. Il s'agit de  $\lambda_3$  dont une des composantes sera changée. On peut, pour cela, décider de cette composante dans le processus ou tirer celle-ci aléatoirement. La 7ème composante sera ici changée.

$$\lambda_3 = 01101010 \rightsquigarrow 011010\mathbf{00} = \lambda_3^*$$

Il est intéressant de noter que  $\lambda_3$  a été muté sans qu'aucune évaluation n'ait été effectuée.

### Nouvelle population (Génération un)

Nous pouvons maintenant examiner la nouvelle population, correspondant à la deuxième génération et réitérer le processus. Réévaluons ces nouveaux individus.

Nouveaux Eléments (renommés)	Eléments Codés	Adaptation : $f(\alpha_i)$
$\lambda_1 \rightsquigarrow \alpha_1$	10111100	0,980225
$\beta_2 \rightsquigarrow \alpha_2$	01101100	0,794678
$\lambda_3^* \rightsquigarrow \alpha_3$	01101000	0,483398
$\beta_4 \rightsquigarrow \alpha_4$	01101100	0,975586
<b>Cumul</b>		<b>3.232</b>

(11)

Lorsqu'on compare les tableaux 10 et 11, plusieurs remarques viennent à l'esprit et méritent d'être notées :

- Le meilleur individu  $\alpha_1$  est un individu nouveau (issu du croisement de  $\theta_2$  et de  $\theta_4$ )
- Cet individu permet d'avoir une adaptation supérieure à celle du meilleur individu de la population originale (0.98 pour  $\alpha_1$  contre 0.97 pour  $\theta_4$ )
- Ce dernier élément est d'ailleurs toujours présent ici (il est ici renommé  $\alpha_4$ )
- L'adaptation totale (et donc l'adaptation moyenne) est supérieure à sa valeur de départ (3.232 contre 2.593 pour la génération zéro).

Chacune des opérations décrites ici ne prend que quelques centièmes de secondes sur un ordinateur, on trouve une valeur de  $x = 0,499959$  après 100 générations et 2,5s de calcul.

Fonctions présentant un intérêt (au moins graphique)

$$y = x^2 \sin(x^{0.5}) \cdot z \ln(\cos(z))^2$$

$$z = x(10 - x) \sin(x^{0.5}) \cdot y \ln(\cos(y))^2$$

## Références

- [1] Alliot J.M. and T. Schiex (1993) : “*Intelligence artificielle et informatique théorique*”, Cépadués Editions.
- [2] Andreoni J. and J. Miller (1995) : “*Auctions with Artificial Adaptive Agents*”, Games and Economic behavior, Vol. 10, pp 39-64.
- [3] Arifovic J. (1994) : “*Genetic algorithm learning and the cobwell model*”, Journal of Economic Dynamic and Control, No 18, pp.2-28.
- [4] Arifovic J. and C. Eaton (1995) : “*Coordination via Genetic Learning*”, Computational Economics, Vol.8, No. 3, pp. 181-203.
- [5] Arthur W. B. (1991) : “*Designing Economic Agents that Act Like Human Agents : A Behavioral Approach to Bounded Rationality*”, AEA Papers and proceedings, Vol. 81, No.2.
- [6] Axelrod R. (1987) : “*The Evolution of Strategies in the Iterated Prisoner’s Dilemma*”, in Genetic Algorithms and Simulated Annealing, L. Davis editor, Pitman, London.
- [7] Beaumont P. and P. Bradshaw (1995) : “*A distributed Parallel Genetic algorithm for solving optimal Growth Models*”, Computational Economics, Vol.8, No. 3, pp. 159-179.
- [8] Birchenhall C. (1995) : “*Introduction of Computational Economics special issue on Genetic algorithm*”, Computational Economics, Vol.8, No. 3, pp. 155-158.
- [9] Birchenhall C. (1995) : “*Modular technical change and Genetic algorithms*”, Computational Economics, Vol.8, No. 3, pp. 233-253.
- [10] Catoni O. (1990) : “*Large deviations for Annealing*”, Thèse de Doctorat, Université de Paris XI..
- [11] Cerf R. (1994) : “*Une théorie asymptotique des algorithmes génétiques*”, These de Doctorat, Université de Montpellier II.
- [12] Dorsey R. E. and W. J. Mayer (1995) : “*Genetic Algorithms for Estimation Problems With Multiple Optima, Nondifferentiability, and Other Irregular Features*”, Journal of Business and Economic Statistics, Vol.13, No 1.
- [13] Durand N., N.Alech, J. M. Alliot and M. Shoenauer (1994) : “*Genetic Algorithms for Optimal Air Traffic Conflict Resolution*” In Proceedings of the second Singapore conference on Intelligent Systems, SPICIS.
- [14] Farley A. M. and S. Jones (1994) : “*Using a Genetic Algorithm to determine an Index of Leading Economic Indicators*”, Computational Economics, Vol. 7, No 3.

- [15] Freidlin M. I. and Wentzell (1983) : “*Random Perturbations of Dynamical Systems*”, Springer Verlag, New-York.
- [16] Goffe L., Ferrier G. D. and J. Roger (1994) : “*Global optimization of statistical functions with simulated annealing*”, Journal of Econometrics, Vol. 60, pp. 65-99.
- [17] Goldberg D. (1989) : “*Genetic Algorithms*” Addison Wesley editor.
- [18] Holland J. H. (1975) : “*Adaptation in Natural and Artificial Systems*” , University of Michigan press.
- [19] Holland J. H. and J. H Miller (1991) : “*Artificial Adaptation Agents in Economic Theory*”, American Economic Review papers Proceedings, Vol. 81, no 2.
- [20] Herrnstein R. J. (1991) : “*Experiments on stable Suboptimality in Individual Behavior*”, AEA Papers and proceedings”, Vol. 81, No.2.
- [21] Michalewicz Z. (1991) : “*Genetic Algorithms + Data Structures = Evolution programs*” Springer Verlag, New-York.
- [22] Petit-Singeot F. and R. Cazoulat (1994) “*Réplicateurs Adaptatifs et Théorie des Jeux*” , in Journées Evolution Artificielle, 20-23 septembre 94, ENAC.
- [23] Tordjman H. (1994) : “*Dynamiques spéculatives, hétérogénéité des agents et apprentissage : Le cas des taux de change*”, These de Doctorat, C.E.F.I.;, Université d’ Aix Marseille II.
- [24] Trouvé A. (1993) : “*Parallélisation massive du recuit simulé*”, Thèse de Doctorat, Université de Paris XI..
- [25] Vriend N. J. (1995) : “*Self-Organisation of Markets : An Example of a Computational Approach*”, Computational Economics, Vol.8, No. 3, pp. 205-231.
- [26] Yin X. and N. Gernay (1993) : “*A Fast Genetic Algorithm with sharing scheme using cluster analysis methods in Multimodal function optimization*”, in Proceedings of the Artificial neural Nets and Genetic algorithms.